ADL & NLPCC 2014 Tutorial
December 6, 2014
ShenZhen China

Semantic Matching in Search

Jun Xu
junxu@ict.ac.cn

Institute of Computing Technology
Chinese Academy of Sciences

People Who Also Contributed to This Tutorial

Hang Li

Outline of Tutorial

- Semantic Matching between Query and Document
- Approaches to Semantic Matching
  1. Matching by Query Reformulation
  2. Matching with Term Dependency Model
  3. Matching with Translation Model
  4. Matching with Topic Model
  5. Matching with Latent Space Model
- Summary

A Good Web Search Engine

- Must be good at
  - Relevance
  - Coverage
  - Freshness
  - Response time
  - User interface
- Relevance is particularly important

Query Document Mismatch Challenge

Table 1.1: Examples of query document mismatch.

query	document	term match	semantic match
seattle best hotel	seattle best hotels	partial	yes
pool schedule	swimming pool schedule	partial	yes
natural logarithm transform	logarithm transform	partial	yes
china kong	china hong kong	partial	no
why are windows so expensive	why are macs so expensive	partial	no

Why Query Document Mismatch Happens?

- Search is still mainly based on term level matching
- Same intent can be represented by different queries (representations)
- Query document mismatch occurs, when searcher and author use different terms (representations) to describe the same concept

Same Search Intent
Different Query Representations

Table 1.2: Queries about "distance between sun and earth".

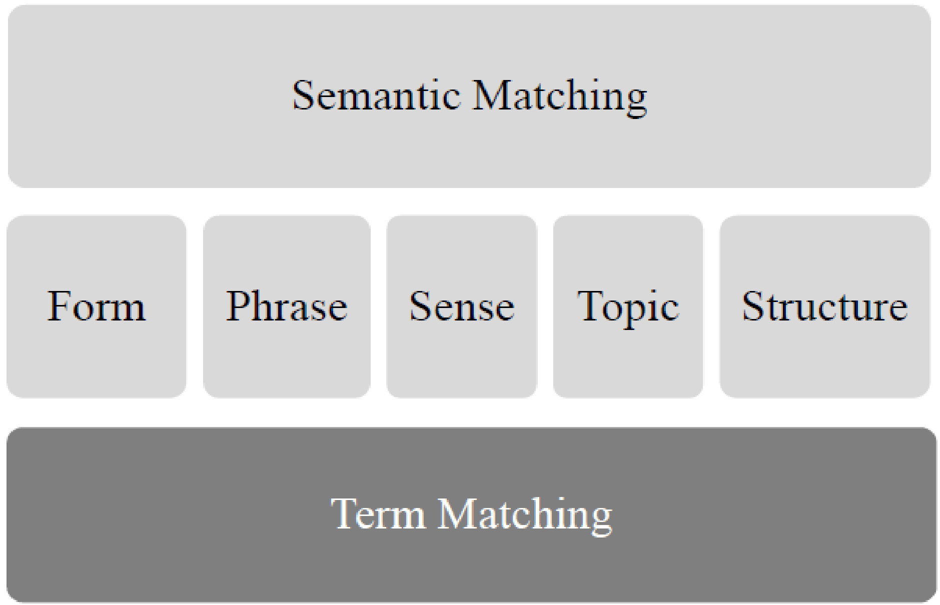
"how far" earth sun	average distance from the earth to the sun
"how far" sun	how far away is the sun from earth
average distance earth sun	average distance from earth to sun
how far from earth to sun	distance from earth to the sun
distance from sun to earth	distance between earth and the sun
distance between earth & sun	distance between earth and sun
how far earth is from the sun	distance from the earth to the sun
distance between earth sun	distance from the sun to the earth
distance of earth from sun	distance from the sun to earth
"how far" sun earth	how far away is the sun from the earth
how far earth from sun	distance between sun and earth
how far from earth is the sun	how far from the earth to the sun
distance from sun to the earth	

Same Search Intent
Different Query Representations

Table 1.3: Queries about "Youtube".

yutube	yuotube	yuo tube
ytube	youtubr	yu tube
youtubo	youtuber	youtubecom
youtube om	youtube music videos	youtube videos
youtube	youtube com	youtube co
youtub com	you tube music videos	yout tube
youtub	you tube com yourtube	your tube
you tube	you tub	you tube video clips
you tube videos	www you tube com	wwww youtube com
www youtube	www youtube com	www youtube co
yotube	www you tube	www utube com
ww youtube com	www utube	www u tube
utube videos	utube com	utube
u tube com	utub	u tube videos
u tube	my tube	toutube
outube	our tube	toutube

Sematic Matching

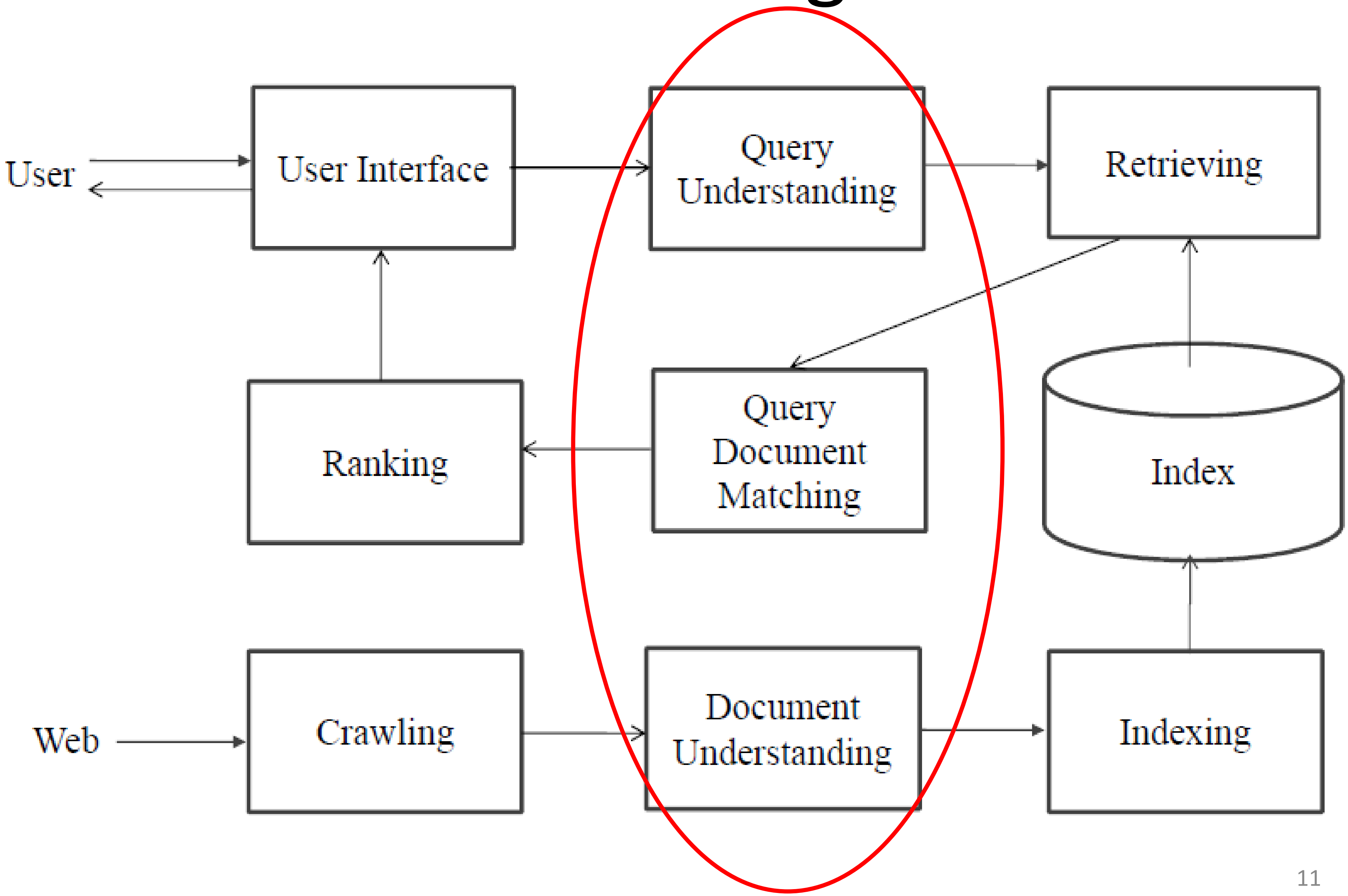


- Reason for mismatch: language understanding by computer is hard, if not impossible
- A more realistic approach: avoid understanding and conduct *matching*

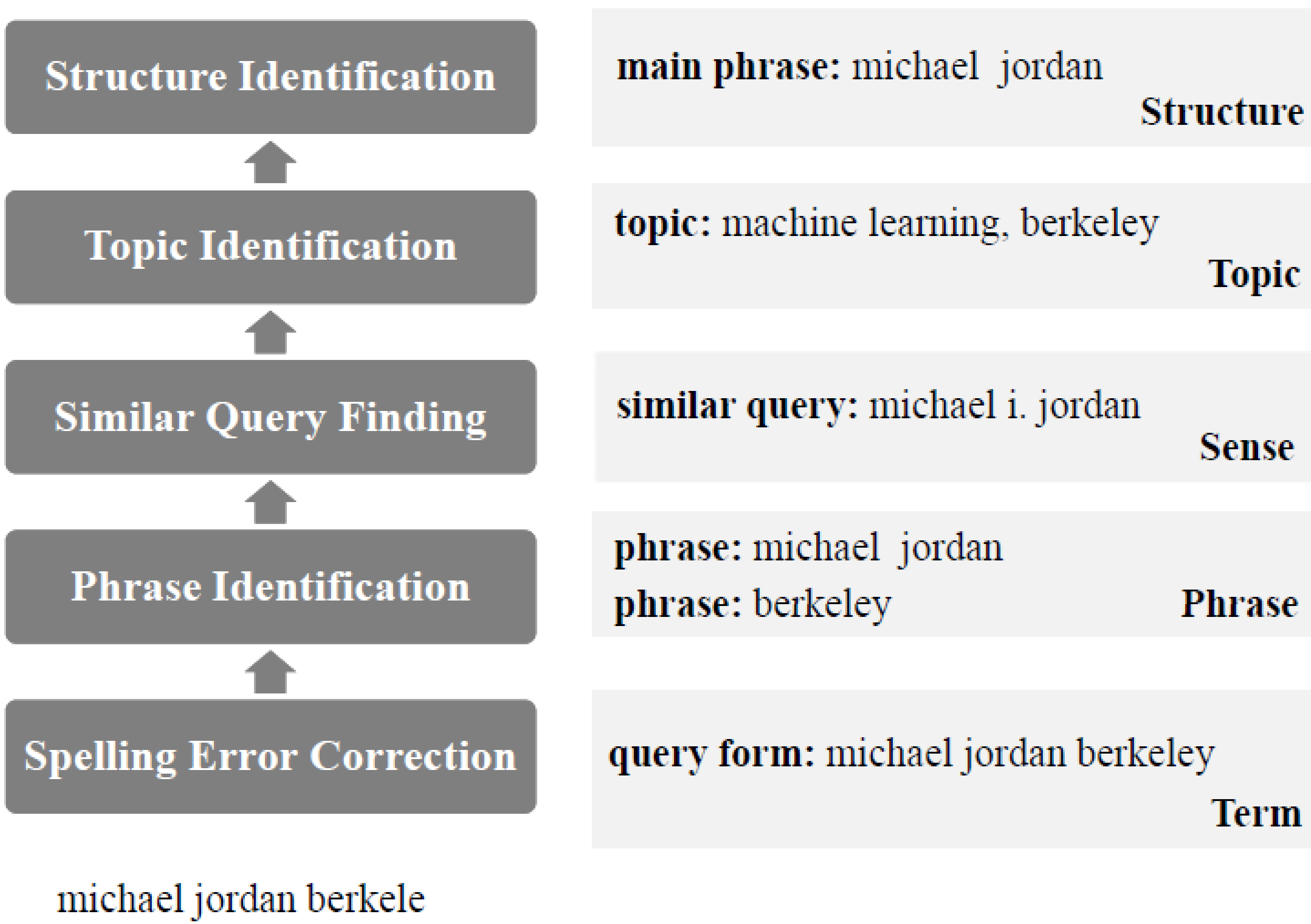
Aspects of Sematic Matching

- More aspects of the query and document can match, more likely the query and document are relevant
  - **Form**: onecar→ onecare
  - **Phrase**: “hot dog” → “hot dog”
  - **Sense**: NY → New York
  - **Topic**: Microsoft Office → Microsoft, PowerPoint, Word, Excel…
  - **Structure**: how far is sun from earth → distance between sun and earth

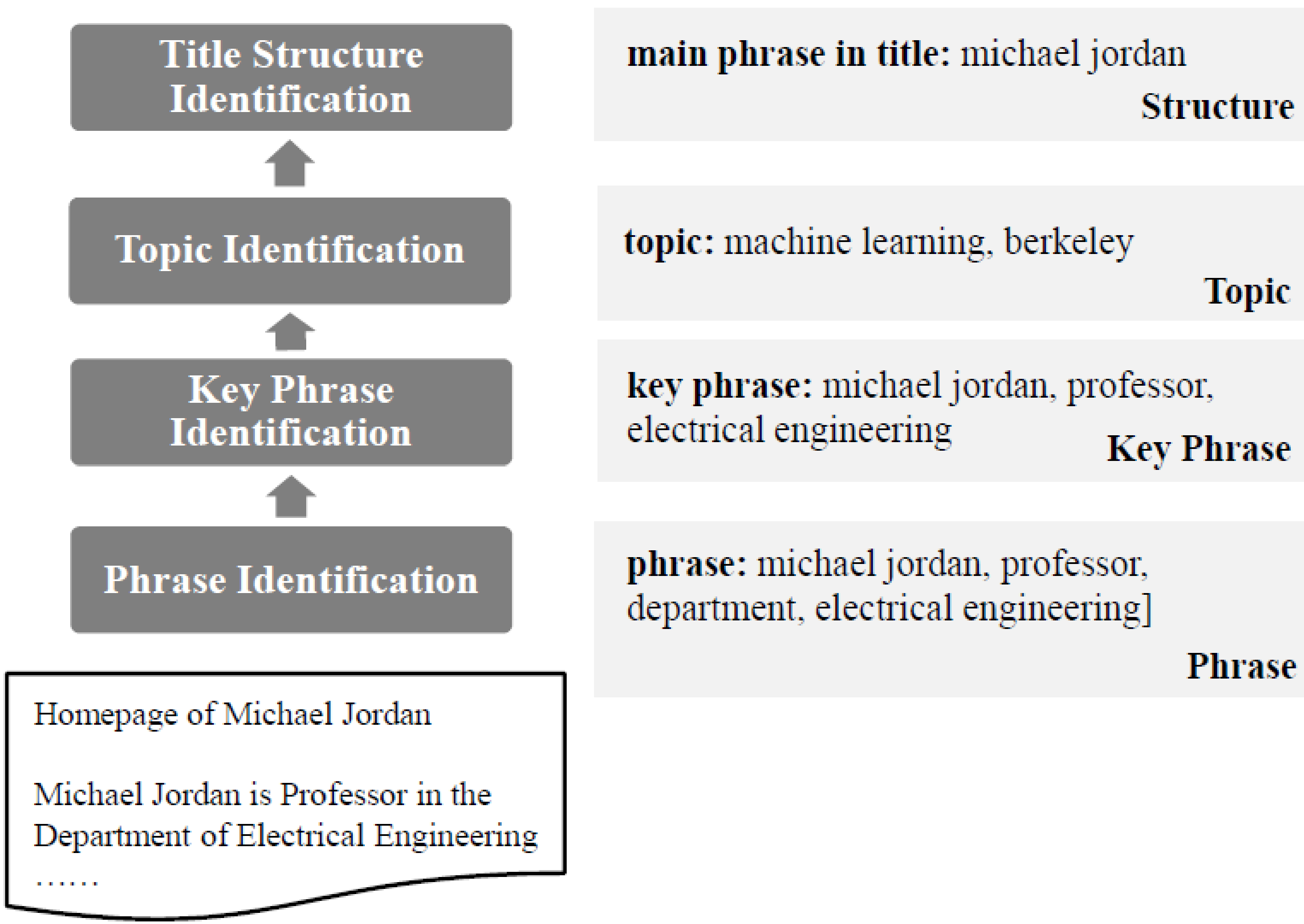
Semantic Matching in Search



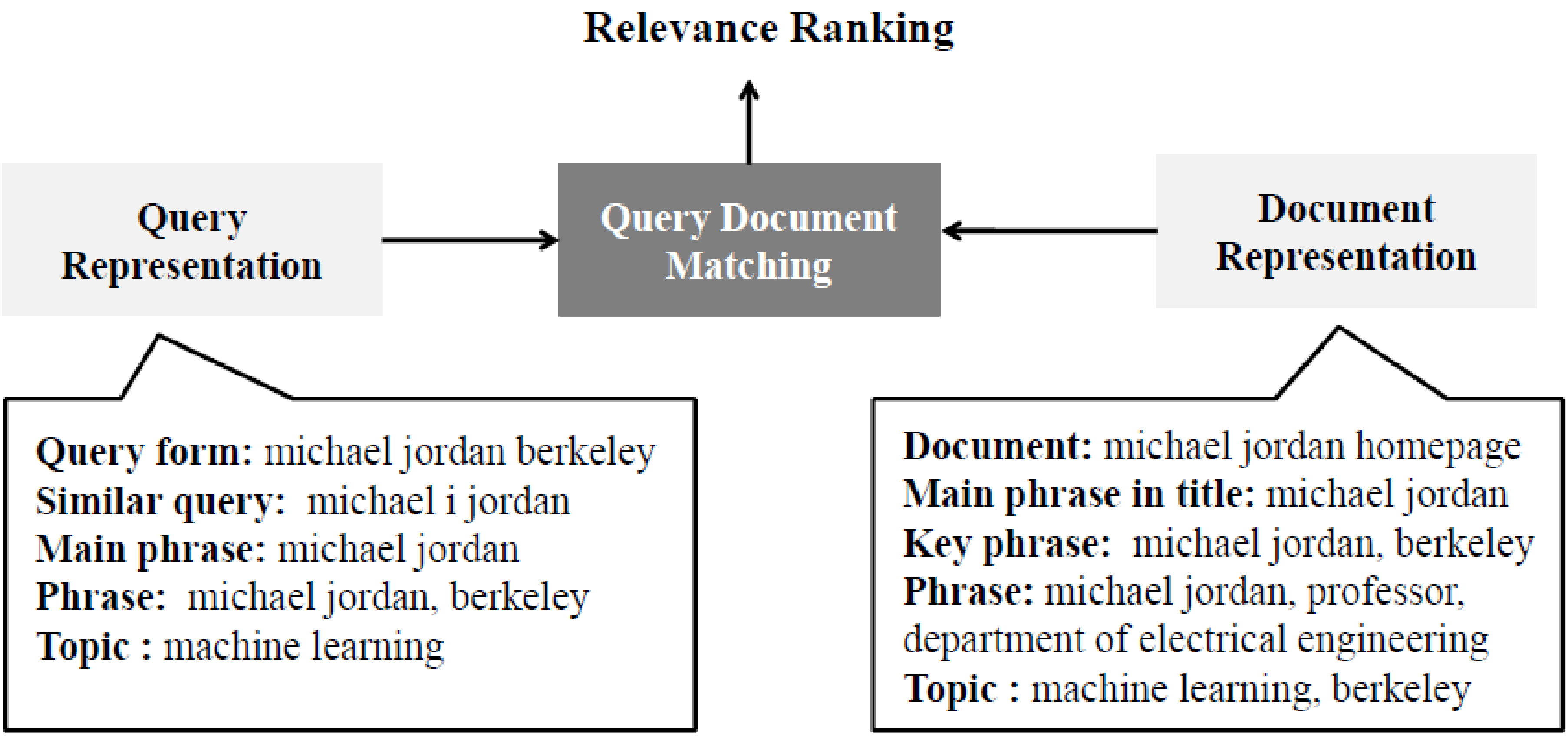
Query Understanding



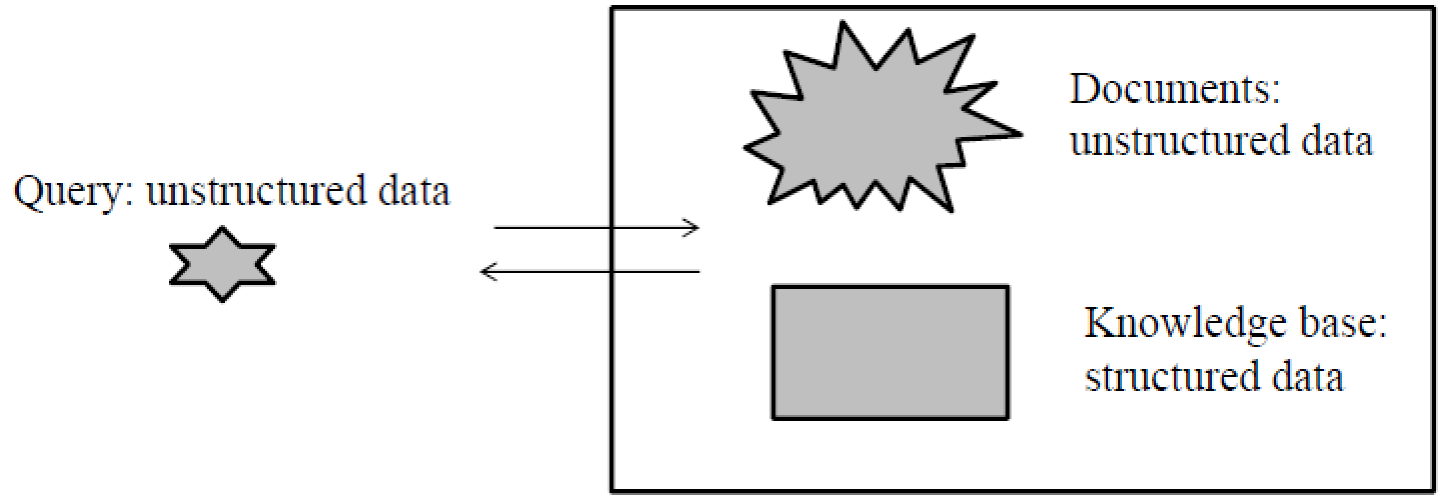
Document Understanding



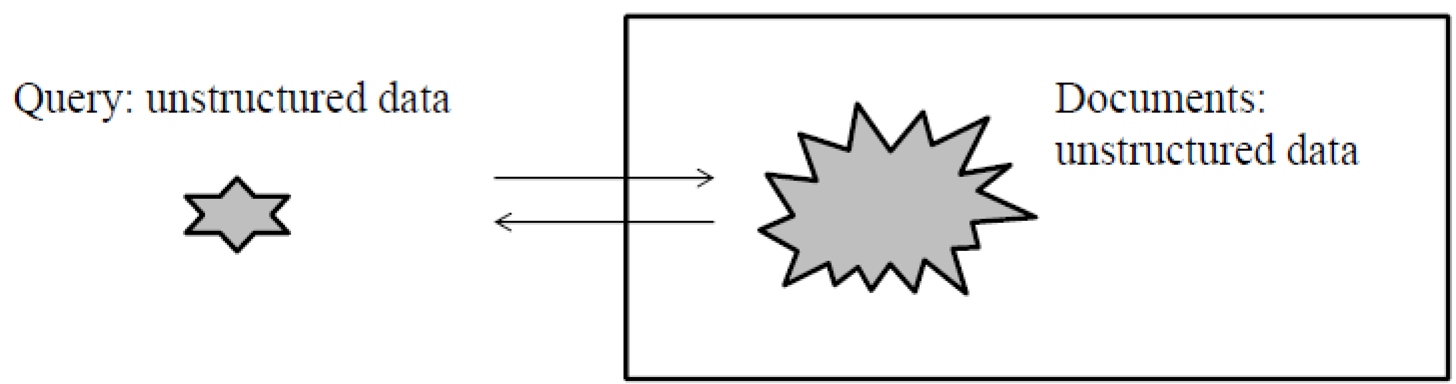
Query Document Matching



Semantic Matching and Semantic Search



Semantic Search



Semantic Matching

Matching and Ranking

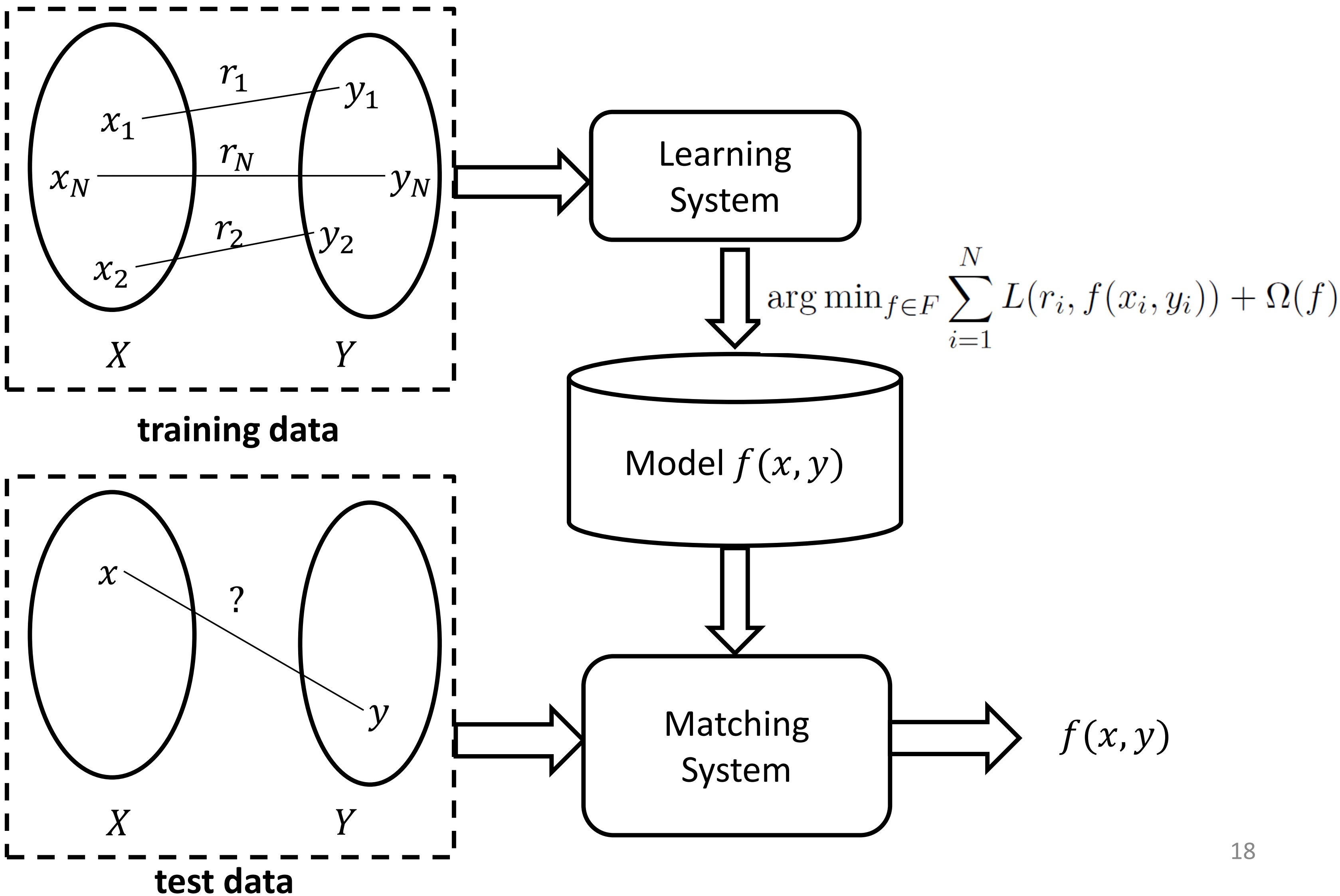
- In search, first matching and then ranking
- Matching results as features for ranking

	Matching	**Ranking**
Prediction	Matching degree between one query and one document	Ranking a list of documents
Model	$f(q, d)$	$f(q, \{d_1, d_2, \cdots, d_N\})$
Challenge	Mismatch	Correct ranking on the top

Semantic Matching in Other Tasks

task	types of texts	relation between texts
search	A=query, B=document	relevance
question answering	A=question, B=answer	answer to question
cross-language IR	A=query, B=document (in diff. lang.)	relevance
short text conversation	A=text, B=text	message and comment
similar document detection	A=text, B=text	similarity
online advertising	A=query, B=ads.	relevance as ads.
paraphrasing	A=sentence, B=sentence	equivalence
textual entailment	A=sentence, B=sentence	entailment

Learning to Match



Challenges

- How to leverage relations in data and prior knowledge
- How to scale up
- How to deal with tail

Approaches to Semantic Matching Between Query and Document

- Matching by Query Reformulation
- Matching with Term Dependency Model
- Matching with Translation Model
- Matching with Topic Model
- Matching with Latent Space Model

Outline of Tutorial

- Semantic Matching between Query and Document
- Approaches to Semantic Matching
  1. Matching by Query Reformulation
  2. Matching with Term Dependency Model
  3. Matching with Translation Model
  4. Matching with Topic Model
  5. Matching with Latent Space Model
- Summary

Query Reformulation

- Transforming the original query to queries (representations) that can better match with documents in the sense of relevance
- Also called
  - Query transformation
  - Query re-writing
  - Query refinement
  - Query alternation

Query Transformation

- Our focus is on how queries can be *transformed* to equivalent, potentially better, queries
  - Queries into paraphrases or "translations"
  - Long queries into shorter queries
  - Short queries into longer queries
  - Queries in one domain to queries in other domains
  - Unstructured queries into structured queries

From Bruce Croft, ECIR 2009

Types of Query Reformulation

type	example
spelling error correction	mlss singapore → miss singapore
merging	face book → facebook
splitting	dataset → data set
stemming	seattle best hotel → seattle best hotels
synonym	ny times → new york times
segmentation	new work times square → "new york" "times square"
query expansion	www → www conference
query deduction	natural logarithm transformation → logarithm transformation
stopword removal\preservation	the new year → "the new year" [1]
paraphrasing	how far is sun from earth → distance between sun and earth

[1] "The new year" is the title of an American movie, and thus the word "the" should not be removed here, although it is usually treated as stopword.

Problems in Query Reformulation

- Query Reformulation
- Similar Query Mining
- Blending

Query Reformulation Problem

- Task
  - Rewrite original query to (multiple) similar queries
- Challenge
  - Topic drift
- Current situation
  - In practice, mainly limited to spelling error correction, query segmentation etc.

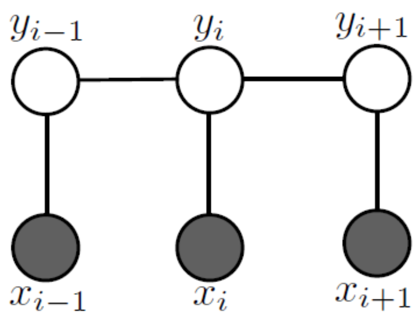
Query Reformulation is Difficult

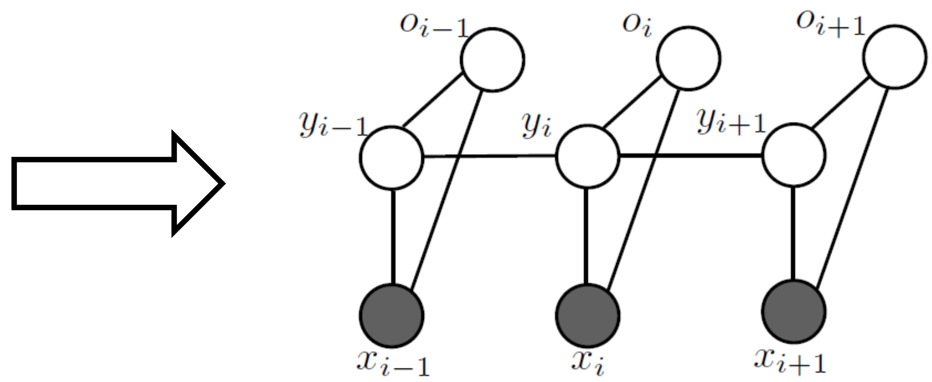
- Depending on the contents of both query and document
- Except
  - Spelling error correction
  - Definite splitting and merging: face book →facebook
  - Definite segmentation: “hot dog”

Methods of Query Reformulation

- Generative approach
  – Source channel model (Brill & Moore, ’00; Cucerzan & Brill, ’04; Duan & Hsu, ‘10)
- Discriminative approach
  – Max entropy (Li et al., ‘06)
  – Log linear model (Okazaki et al., ’08; Wang et al., ‘11)
  – Conditional Random Fields (Guo et al., ‘08)

Conditional Random Field for Query Reformulation (Guo et al., '08)





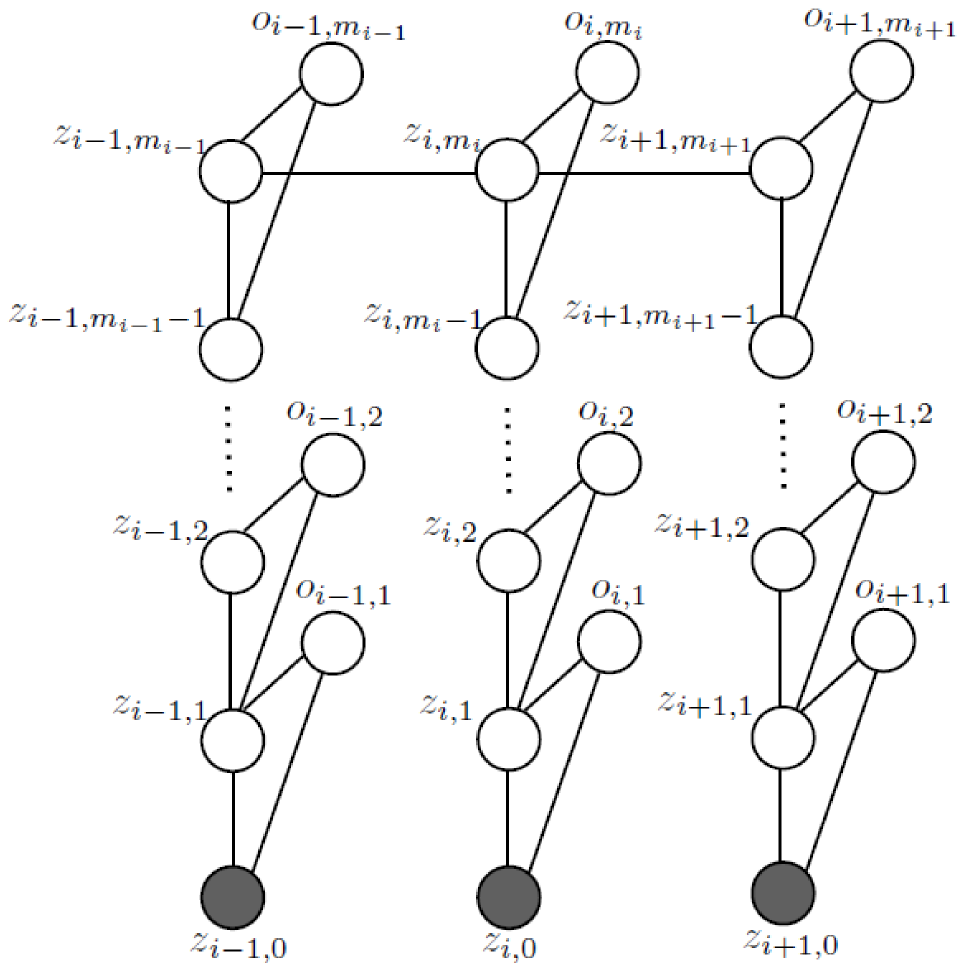
$$\Pr(\boldsymbol{y}, \boldsymbol{o} | \boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{i=1}^{n} \phi(y_{i-1}, y_i) \phi(y_i, o_i, \boldsymbol{x})$$

- $\boldsymbol{x}$: observed noisy query, e.g., window onecar
- $\boldsymbol{y}$: reformulated query, e.g., windows onecare
- $\boldsymbol{o}$: a sequence of operations
- Learning: $P(\boldsymbol{y}, \boldsymbol{o}|\boldsymbol{x})$
- Prediction: $\text{argmax}_{\boldsymbol{y},\boldsymbol{o}} P(\boldsymbol{y}, \boldsymbol{o}|\boldsymbol{x})$

Operations

Task	Operation	Description
Spelling Correction	Deletion	Delete a letter in the word
	Insertion	Insert a letter into the word
	Substitution	Replace a letter in the word with another letter
	Transposition	Switch two letters in the word
Word Splitting	Splitting	Split one word into two words
Word Merging	Merging	Merge two words into one word
Phrase Segmentation	Begin	Mark a word as beginning of phrase
	Middle	Mark a word as middle of phrase
	End	Mark a word as end of phrase
	Out	Mark a word as out of phrase
Word Stemming	+s/-s	Add or Remove suffix '-s'
	+ed/-ed	Add or Remove suffix '-ed'
	+ing/-ing	Add or Remove suffix '-ing'
Acronym Expansion	Expansion	Expand acronym

Extended Model



$$\Pr(\boldsymbol{y}, \vec{\boldsymbol{o}}, \vec{\boldsymbol{z}} | \boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})} \prod_{i=1}^{n} (\phi(y_{i-1}, y_i) \prod_{j_i=1}^{m_i} \phi(z_{i,j_i}, | o_{i,j_i}, z_{i,j_i-1}))$$

Experimental Results

	Pre.	Rec.	F1	Acc.
CRF-QR	**54.48**	**40.75**	**46.63**	**56.27**
Cascaded1	53.38	39.71	45.54	55.57
Cascaded2	53.38	39.71	45.54	55.57
Cascaded3	53.38	39.71	45.54	55.57
Cascaded4	53.45	39.76	45.60	55.60
Cascaded5	53.45	39.76	45.60	55.60
Cascaded6	53.45	39.76	45.60	55.60
Generative	30.46	32.95	31.66	39.10

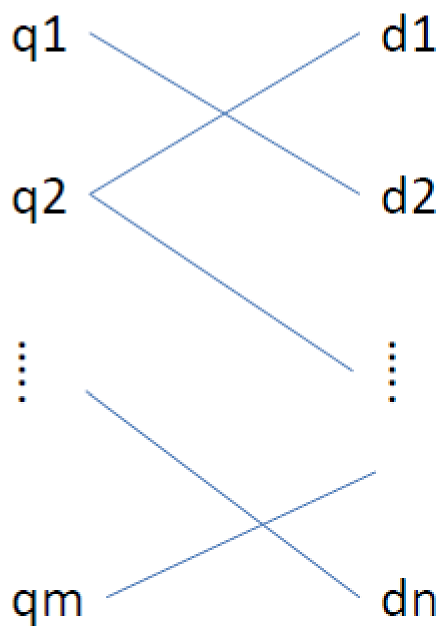
- Data: 10,000 queries, 6,421 queries were refined by human annotators
- Result: extended CRF-QR model significantly outperformed the baselines

Similar Query Mining

- Task
  - Given click-through data for search session data
  - Find similar queries or similar query patterns
    E.g., ny → new York; distance tween X and Y → how far is X from Y
- Challenge
  - Dealing with noise

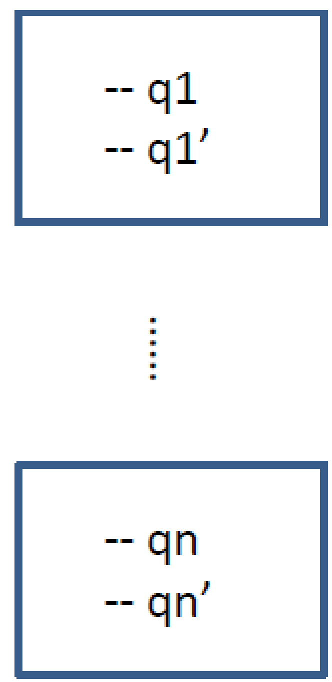
Mining of Similar Queries

Click-through data



Similar queries can be found by co-click

Search session data



Similar queries can be found from users' query reformulations

Methods of Similar Query Mining

- Using click-through data
  - Pearson correlation coefficient (Xu & Xu, ‘11)
  - Agglomerative clustering (Beeferman & Burger, ’00), DBScan (Wen et al., ’01), K-means (Baeza-Yates et al., ‘04), Query stream clustering (Cao et al., ’08; Liao et al., ‘12)
- Using search session data
  - Jacaard similarity (Huang et al., ’03), likelihood ratio (Jones et al., ‘06)
- Learning of query reformulation patterns
  - Mining natural language question patterns (Xue et al., ‘12)
- Learning of query similarity
  - Query similarity as metric learning (Xu & Xu ‘11)

Query Similarity as Metric Learning (Xu & Xu, ’11)

- Given similar query pairs and dissimilar query pairs
- Learn from head queries and *propagate to tail queries*
- Objective function:

$$\max_{M \succeq 0} \sum_{(q_i, q_j) \in S_+} \frac{\phi(q_i)^T M \phi(q_j)}{\sqrt{\phi(q_i)^T M \phi(q_i)} \sqrt{\phi(q_j)^T M \phi(q_j)}}$$

$$- \sum_{(q_i, q_j) \in S_-} \frac{\phi(q_i)^T M \phi(q_j)}{\sqrt{\phi(q_i)^T M \phi(q_i)} \sqrt{\phi(q_j)^T M \phi(q_j)}} - \lambda \|M\|_1$$

Query Similarity as Metric Learning

- $\phi(q)$: N-gram vector space

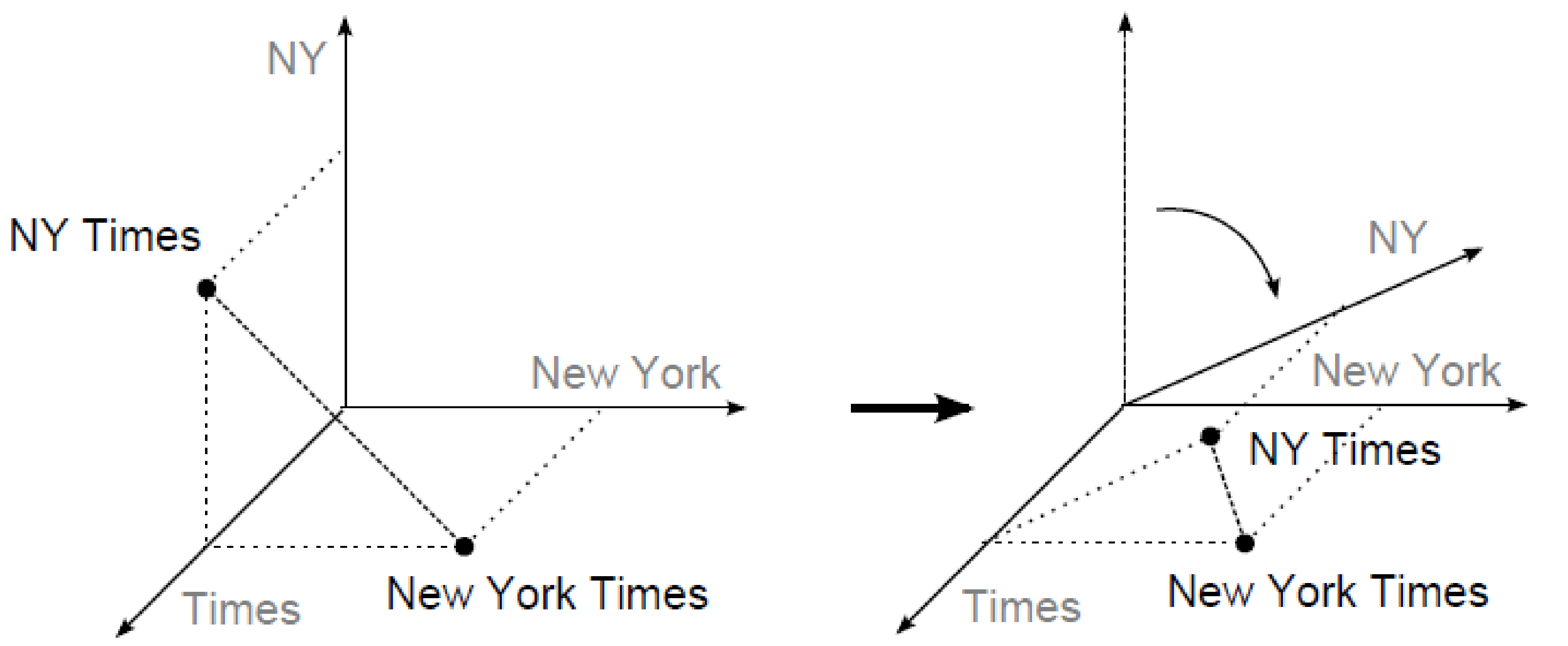
Query	Vectors in n-gram vector space						
	(ny,	new,	york,	times,	ny times,	new york,	...)
NY times	(1,	0,	0,	1,	1,	0,	...)
New York times	(0,	1,	1,	1,	0,	1,	...)

- Learned similarity function (M is positive semi-definite)

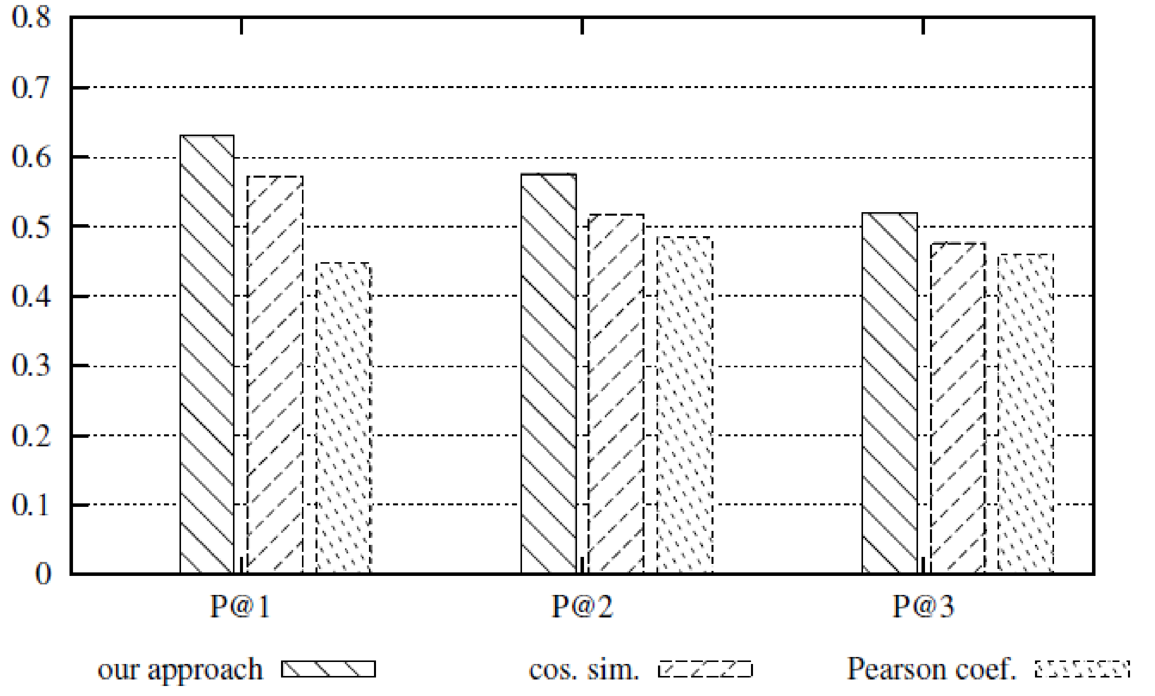
$$sim(\phi(q_i), \phi(q_i)) = \frac{\phi(q_i)^T M \phi(q_j)}{\sqrt{\phi(q_i)^T M \phi(q_i)}\sqrt{\phi(q_j)^T M \phi(q_j)}}$$

Query Similarity as Metric Learning

- Interpretation: transformation between n-gram spaces



Experimental Results



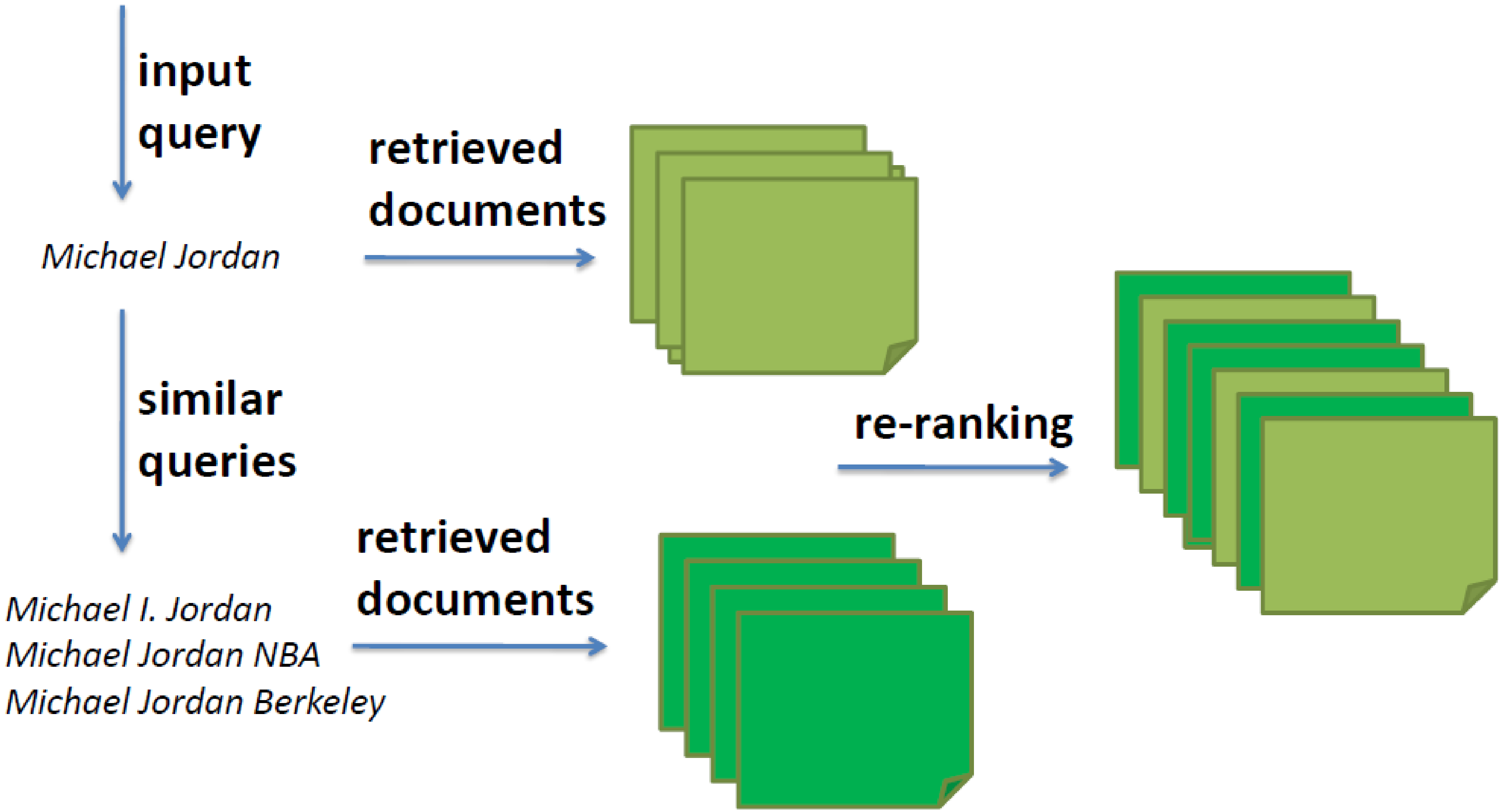
Precision of similar query calculation methods on rare query

- Constantly outperforms the two baselines on rare queries

Blending Problem

- Steps
  - Rewrite original query to multiple similar queries
  - Retrieve with multiple queries
  - Blend results from multiple queries
- Challenges
  - System to sustain searches with multiple queries
  - Blending model: matching scores are not comparable across queries

Blending


input
query
Michael Jordan
retrieved
documents
similar
queries
Michael I. Jordan
Michael Jordan NBA
Michael Jordan Berkeley
retrieved
documents
re-ranking

Methods of Blending

- Linear combination (Xue et al., ‘08)
- Learning to rank (Sheldon et al., ‘11)
- Kernel methods (Wu et al., ‘11)

Kernel Method for Blending (Wu et al., ’11)

- Given query similarity and document similarity
- “Smoothing query and document similarity” by those of similar queries and documents
- Interpretation: nearest neighbor in space of query and document pair (double KNN)
- Automatically learning the weights of combination from click-data

Learning of Matching Model

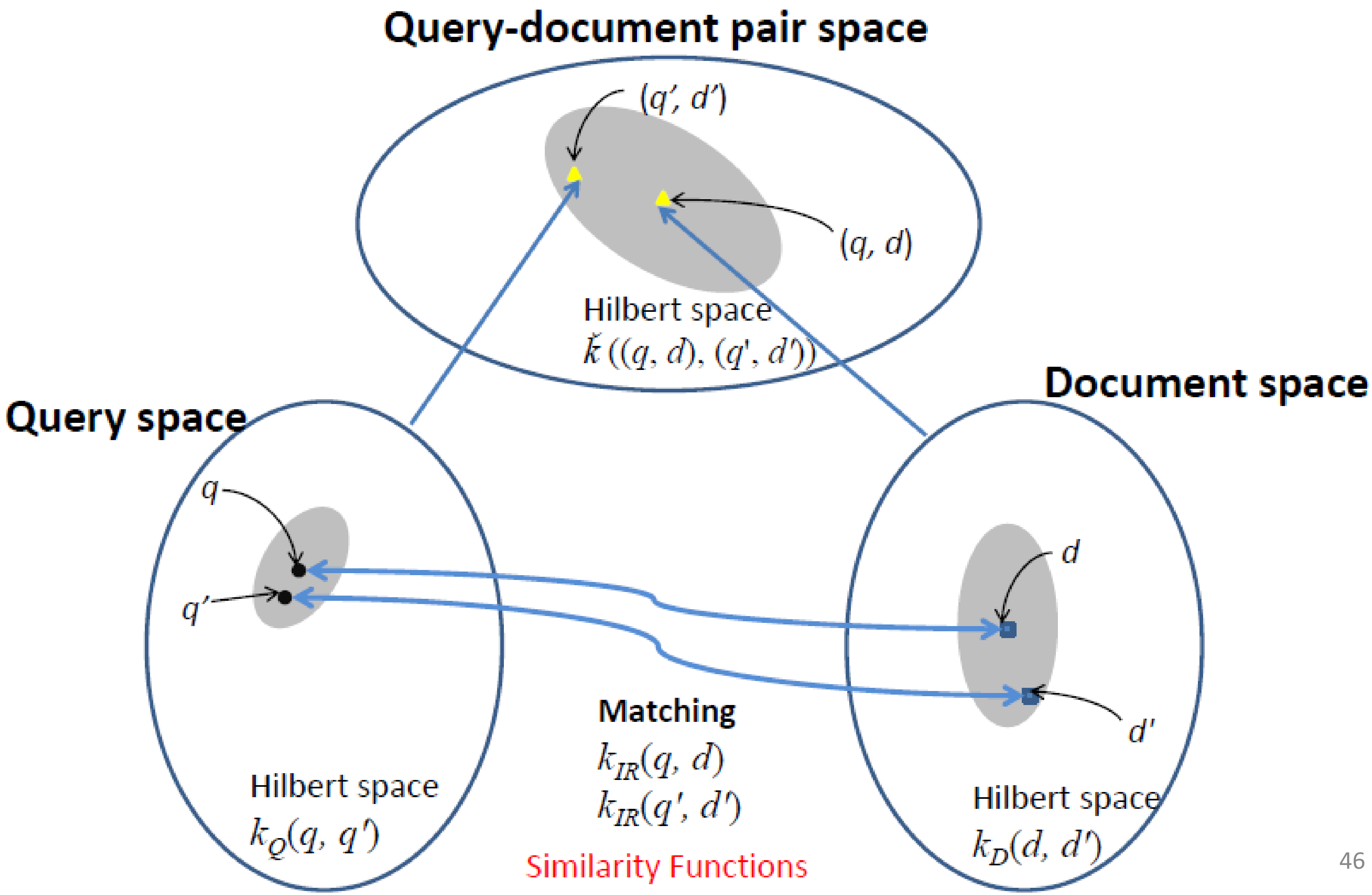
- Matching function: $k(x, y) = \langle \varphi_X(x), \varphi_Y(y) \rangle_{\mathcal{H}}$
- Input: training data $S = \{(x_i, y_i), r_i\}_{1 \le i \le N}$
- Output: matching function
- Optimization

$$\min_{k \in \mathcal{K}} \frac{1}{N} \sum_{i=1}^{N} l(k(x_i, y_i), r_i) + \Omega(k)$$

Learning of Matching Model Using Kernel Method

- Assumption: space of matching functions is RKHS generated by positive definite kernel $\bar{k}$: $(X \times$

Kernel Method



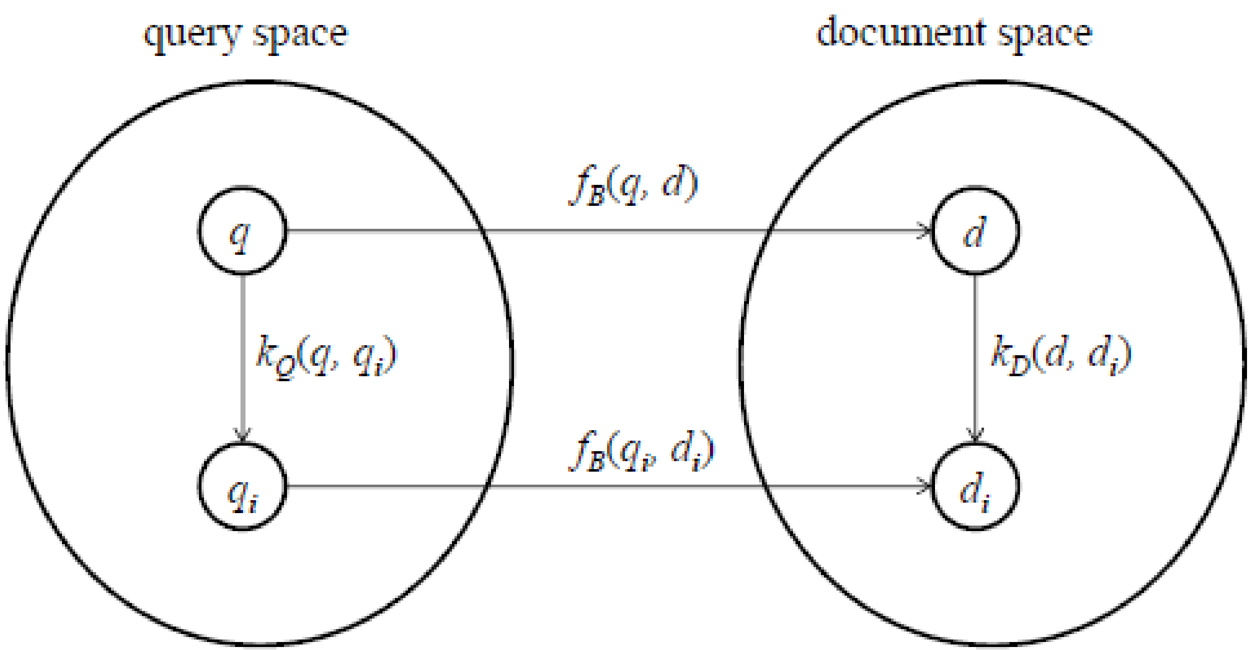
Implementation: Learning of BM25

- BM25: similarity function between query and document, denoted as k_{BM25}
- Kernel:

$$\bar{k}\big((q,d),(q',d')\big) = k_{BM25}(q,d)k_Q(q,q')k_D(d,d')k_{BM25}(q',d')$$

- Solution (called Robust BM25)

$$k_{RBM25} = k_{BM25}(q,d)\sum_{i=1}^{N}\alpha_i k_Q(q,q_i)k_D(d,d_i)k_{BM25}(q_i,d_i)$$



Experimental Results

		MAP	NDCG@1	NDCG@3	NDCG@5
Web search	Robust BM25	0.1192	0.2480	0.2587	0.2716
	Pairwise Kernel	0.1123	0.2241	0.2418	0.2560
	Query Expansion	0.0963	0.1797	0.2061	0.2237
	BM25	0.0908	0.1728	0.2019	0.2180
Enterprise search	Robust BM25	0.3122	0.4780	0.5065	0.5295
	Pairwise Kernel	0.2766	0.4465	0.4769	0.4971
	Query Expansion	0.2755	0.4076	0.4712	0.4958
	BM25	0.2745	0.4246	0.4531	0.4741

- Robust BM25 significantly outperforms the baselines, in terms of all measures on both data sets

References

- Ricardo Baeza-Yates, Carlos Hurtado, and Marcelo Mendoza. Query clustering for boosting web page ranking. Advances in Web Intelligence, volume 3034 of Lecture Notes in Computer Science, pages 164–175. Springer Berlin Heidelberg, 2004.
- Doug Beeferman and Adam Berger. Agglomerative clustering of a search engine query log. In KDD ’00, pages 407–416, 2000.
- Eric Brill and Robert C. Moore. An improved error model for noisy channel spelling correction. In ACL '00, pages 286-293, 2000.
- Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. Selecting good expansion terms for pseudo-relevance feedback. In SIGIR ’08, pages 243–250, 2008.
- Silviu Cucerzan, Eric Brill. Spelling Correction as an Iterative Process that Exploits the Collective Knowledge of Web Users. In EMNLP 2004, pages 293–300, 2004.
- Huizhong Duan and Bo-June (Paul) Hsu. 2011. Online spelling correction for query completion. In WWW '11, pages 117-126, 2011.
- Chien-Kang Huang, Lee-Feng Chien, and Yen-Jen Oyang. Relevant term suggestion in interactive web search based on contextual information in query session logs. J. Am. Soc. Inf. Sci. Technol., 54(7):638–649, May 2003.
- Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. Generating query substitutions. In WWW ’06, pages 387–396, 2006.

References

- Mu Li, Yang Zhang, Muhua Zhu, and Ming Zhou. Exploring distributional similarity based models for query spelling correction. In ACL ‘06, pages 1025-1032. 2006.
- Zhen Liao, Daxin Jiang, Enhong Chen, Jian Pei, Huanhuan Cao, and Hang Li. Mining concept sequences from large-scale search logs for context-aware query suggestion. ACM Trans. Intell. Syst. Technol., 3(1):17:1–17:40, October 2011.
- Naoaki Okazaki, Yoshimasa Tsuruoka, Sophia Ananiadou, and Jun'ichi Tsujii. A discriminative candidate generator for string transformations. In EMNLP '08, pages 447-456, 2008.
- Daniel Sheldon, Milad Shokouhi, Martin Szummer, and Nick Craswell.Lambdamerge: Merging the results of query reformulations. In WSDM ’11, pages 795–804, 2011.
- Ziqi Wang, Gu Xu, Hang Li and Ming Zhang, A Fast and Accurate Method for Approximate String Search, In ACL-HLT’11, pages 52-61, 2011.
- Ji-Rong Wen, Jian-Yun Nie, and Hong-Jiang Zhang. Clustering user queries of a search engine. In WWW ’01, pages 162–168, 2001.
- Wei Wu, Jun Xu, Hang Li, and Satoshi Oyama. Learning a robust relevance model for search using kernel methods. J. Mach. Learn. Res., 12:1429–1458, July 2011
- Jingfang Xu and Gu Xu. Learning similarity function for rare queries. In WSDM ’11, pages 615–624, 2011.
- Xiaobing Xue, Yu Tao, Daxin Jiang, and Hang Li. Automatically mining question reformulation patterns from search log data. In ACL ’12, pages 187–192, 2012.

Coffee Break

Outline of Tutorial

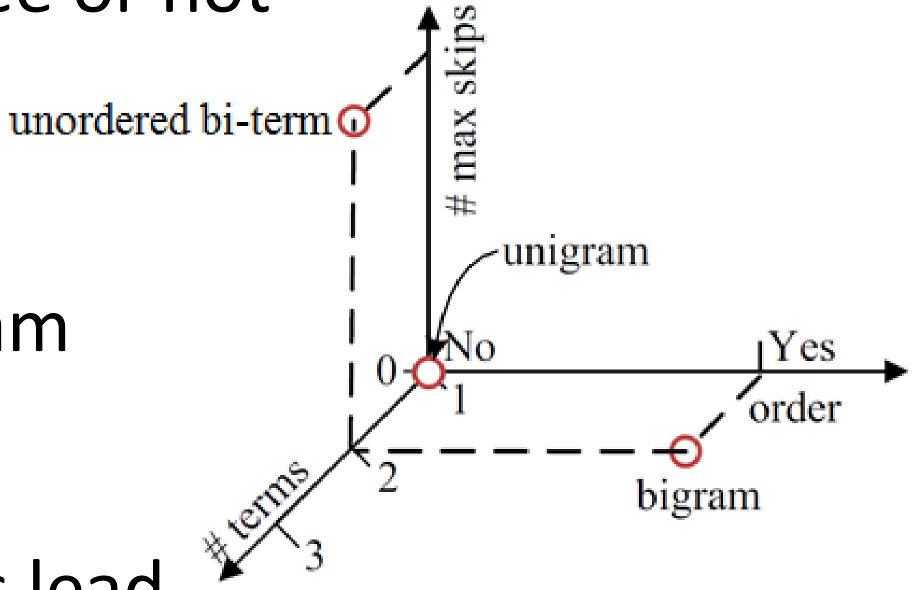
- Semantic Matching between Query and Document
- Approaches to Semantic Matching
  1. Matching by Query Reformulation
  2. Matching with Term Dependency Model
  3. Matching with Translation Model
  4. Matching with Topic Model
  5. Matching with Latent Space Model
- Summary

Matching based on Term Dependency

- Matching of consecutive terms in query and document indicates higher relevance
  - “hot dog”
  - “hot dog” ≠ hot + dog
- Query: order is quite free, but not completely free
  - “hot dog recipe”, “recipe hot dog”
  - “hot recipe dog” ×
- Term dependency: a sequence of terms representing *soft* query segmentation

Factors of Term Dependency

- # terms: number of terms in n-gram
  - 1 term (unigram)
  - Multiple terms (bigram, bi-terms …)
- Order: order of terms is free or not
  - N-gram
  - Unordered N-terms
- Skip: maximum number of terms skipped within n-gram
  - No skip
  - S skips
- Different choices of factors lead to different types of term dependencies



Types of Term Dependency

- Term dependency in query
  - Noun phrases (Bendersky & Croft, ’08)
  - Phrases & proximities (Bendersky & Croft, ’10; Shi & Nie, ’10; Bendersky & Croft, ‘12)
- Latent term dependency
  - Pseudo relevance feedback (Cao et al., ’08; Metzler & Croft ’07; Lease ’08; Bendersky et al., ’11)
  - Query expansion (Metzler ’11)

Addressing Term Mismatch based on Term Dependency

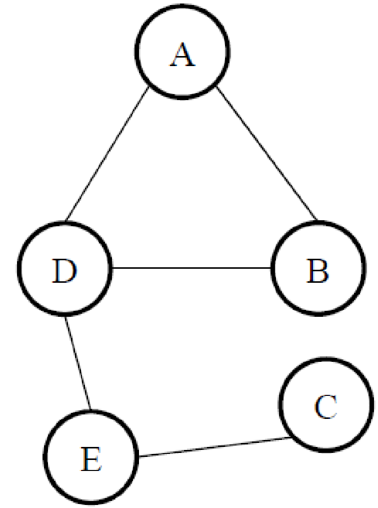
- Explicit term dependency represents degree of matching between query and document
  - Document including “hot dog” has higher matching degree than document including “hot” and “dog”
- Latent term dependency uses relations with additional terms to help ‘infer’ degree of matching
  - Query “airplane” has nonzero matching score with document including “aircraft”

Methods of Matching with Term Dependencies

- Term dependencies using Markov Random Fields (MRF)
  - Explicit term dependencies (Metzler & Croft, ’05)
  - Latent term dependencies (Metzler & Croft, 2008; Bendersky et al, ’11)
  - Weighted term dependencies (Bendersky et al., ’10; Bendersky et al, ’11)
- Extended IR models (Bendersky & Croft, ’12; Shi & Nie, ’10)

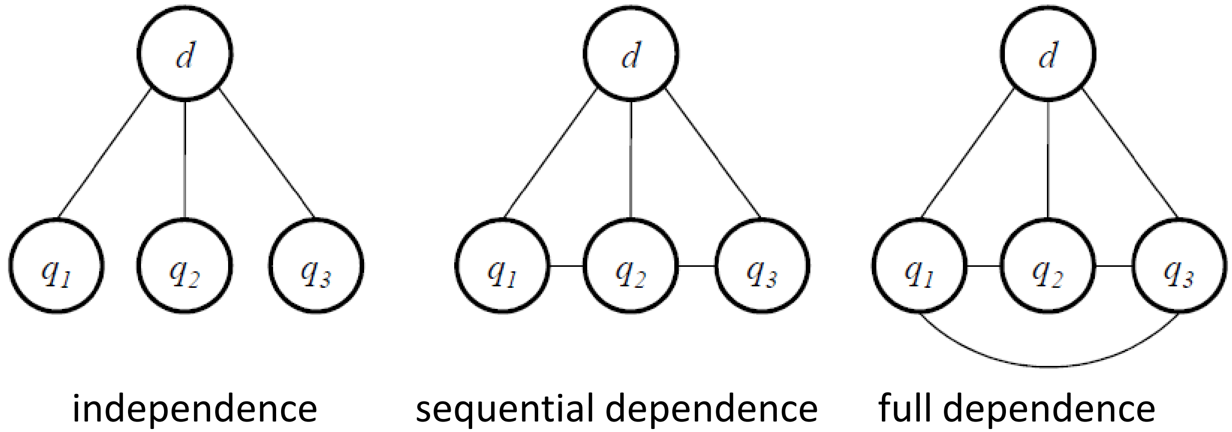
Markov Random Fields (MRF)

- Joint probability distribution represented by an undirected graph
  - Nodes: random variables
  - Edges: probabilistic dependencies
  - Cliques: subset of nodes such that every two nodes are connected
- Factorization of joint probability based on cliques



$$P(x_1, \cdots, x_N) = \underbrace{\frac{1}{Z}}_{\text{normalizing factor}} \prod_{c \in clique(G)} \underbrace{\psi(c)}_{\text{potential function}}$$

Modeling Term Dependencies with MRF (Metzler & Croft, 2005)



- Nodes
  - Document node
  - One node for each query term
- Edges
  - Each query node is linked with document node
  - Dependent terms are linked together

Modeling Term Dependencies with MRF

- Cliques
  - Representing how query terms are matched in document
  - Matching scores determined by potential function
- Joint probability

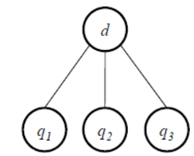
$$P(\mathbf{q}, \mathbf{d}) = \frac{1}{Z} \prod_{c \in clique(G)} \exp(\lambda_c f(c))$$

- Matching function

$$F(q, d) = \sum_{c \in \mathrm{clique(G)}} \lambda_c f(c)$$

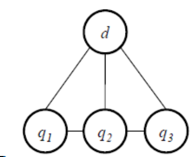
Modeling Term Dependencies with MRF

- Three types of feature functions $f(c)$
  - Fully independent



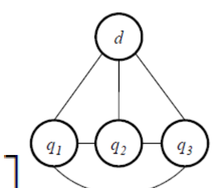
$$f_1(q_i, d) = \log \left[(1-\alpha) \frac{tf(q_i, d)}{|d|} + \alpha \frac{cf(q_i)}{|C|} \right]$$

  - Sequentially dependent



$$f_2(q_i, \cdots, q_{i+k}, d) = \log \left[(1-\alpha) \frac{tf(q_i, \cdots, q_{i+k}, d)}{|d|} + \alpha \frac{cf(q_i, \cdots, q_{i+k})}{|C|} \right]$$

  - Fully dependent



$$f_3(q_i, \cdots, q_j, d) = \log \left[(1-\alpha) \frac{tf(q_i, \cdots, q_j, d)}{|d|} + \alpha \frac{cf(q_i, \cdots, q_j)}{|C|} \right]$$

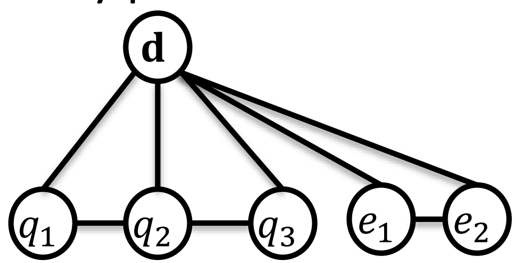
Experimental Results

	fully independent		sequentially dependent		fully dependent	
	MAP	P@10	MAP	P@10	MAP	P@10
AP	0.1775	0.2912	0.1867*	0.2980	0.1866*	0.3068*
WSJ	0.2592	0.4327	0.2776*	0.4427	0.2738*	0.4413
WT10g	0.2032	0.2866	0.2167*	0.2948	0.2231*	0.3031
GOV2	0.2502	0.4837	0.2832*	0.5714*	0.2844*	0.5837*

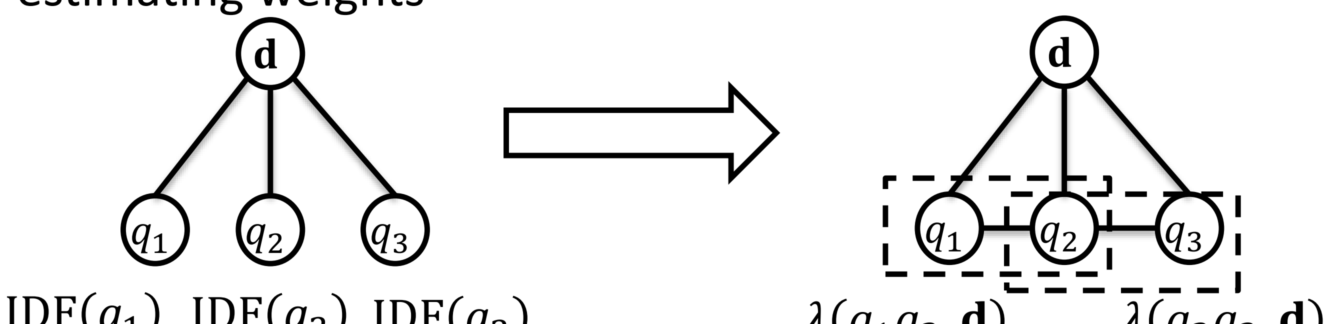
- Sequentially dependent and fully dependent outperform the baseline of fully independent

MRF Extensions

- Latent Term Dependencies(Metzler & Croft, 2007)
  - Latent terms exist behind query
  - E.g., collecting terms by pseudo relevance feedback



- Weighted Term Dependencies (Bendersky et al., 2010)
  - High weights for most discriminative term dependencies (like IDF for unigram)
  - Leveraging different data resources such as web N-gram, Wikipedia etc. for estimating weights



Extended IR Model

- IR model as asymmetric kernels (Xu et al., '10)

$$\text{BM25-Kernel}(q,d) = \sum_t \text{BM25-Kernel}_t(q,d)$$

$$\text{BM25-Kernel}_t(q,d) = \sum_x \text{IDF}_t(x) \times \frac{(k_3+1)\times f_t(x,q)}{k_3 + f_t(x,q)} \times \frac{(k_1+1)\times f_t(x,d)}{k_1\left(1-b+b\frac{f_t(d)}{avgf_t}\right)+f_t(x,d)}$$

- Dependency language model (Gao et al., '04)
  - Generate linkage l according to $P(l|d)$
  - Generate q according to $P(q|l,d)$

$$P(q|d) = \sum_l P(q,l|d) = \sum_l P(l|d)P(q|l,d)$$

References

- Michael Bendersky and W. Bruce Croft. Discovering Key Concepts in Verbose Queries. In Proc. of SIGIR 2008.
- Michael Bendersky, Donald Metzler, and W. Bruce Croft. Learning Concept Importance using a Weighted Dependence Model. In Proc. of WSDM 2010.
- Michael Bendersky, Donald Metzler, and W. Bruce Croft. Parameterized Concept Weighting in Verbose Queries. In Proc. of SIGIR 2011.
- Michael Bendersky and W. Bruce Croft. Modeling higher-order Term Dependencies in Information Retrieval using Query Hypergrahics. In Proc. of SIGIR 2012.
- Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson: Selecting Good Expansion Terms for Pseudo-Relevance Feedback. In Proc. of SIGIR 2008.
- Van Dang, Michael Bendersky, and W. Bruce Croft. Learning to Rank Query Reformulations. In Proc. of SIGIR 2010.
- Hao Lang, Donald Metzler, Bin Wang, and Jin-Tao Li. Improved Latent Concept Expansion using Hierarchical Markov Random Fields. In Proc. of CIKM 2010.
- Matthew Lease, James Allan, and Bruce Croft. Regression Rank: Learning to Meet the Opportunity of Descriptive Queries. In Proc. of ECIR 2009.
- Matthew Lease. Incorporating Relevance and Pseudo-relevance Feedback in the Markov Random Field Model. In Proc. of TREC 2008.

References

- Matthew Lease. An Improved Markov Random Field Model for Supporting Verbose Queries. In Proc. of SIGIR 2009.
- Donald Metzler and W. Bruce Croft. A Markov Random Field Model for Term Dependencies. In Proc. of SIGIR 2005.
- Donald Metzler and W. Bruce Croft. Linear Feature-based Models for Information Retrieval. Information Retrieval, 2006.
- Donald Metzler and W. Bruce Croft. Latent Concept Expansion using Markov Random Fields. In Proc. of SIGIR 2008.
- Donald Metzler. Feature-based Query Expansion. In Proc. of SIGIR 2011.
- Lixin Shi and Jian-Yun Nie. Using Various Term dependencies according to Their Utilities. In Proc. of CIKM 2010.
- Krysta M. Svore, Pallika H. Kanani, and Nazan Khan. How Good is a Span of Terms? Exploiting Proximity to Improve Web Retrieval. In Proc. of SIGIR 2010.
- Lidan Wang, Donald Metzler, and Jimmy Lin. Ranking under Temporal Constraints. In Proc. of CIKM 2010.
- Jun Xu, Hang Li, and Chaoliang Zhong. Relevance Ranking using Kernels. In Proc. of AIRS 2010.
- Le Zhao and Jamie Callan. Term Necessity Prediction. In Proc. of CIKM 2010.

Outline of Tutorial

- Semantic Matching between Query and Document
- Approaches to Semantic Matching
  1. Matching by Query Reformulation
  2. Matching with Term Dependency Model
  3. Matching with Translation Model
  4. Matching with Topic Model
  5. Matching with Latent Space Model
- Summary

Outline

- Statistical Machine Translation
- Search as Translation
- Methods for Matching with Translation Models

Statistical Machine Translation (SMT)

- Given sentence $C = c_1 c_2 \cdots c_J$ in source language, translates it into sentence $E = e_1 e_2 \cdots e_I$ in target language

$$
\begin{aligned}
E^* &= \arg\max_E P(E|C) \\
&= \arg\max_E \frac{P(C|E)P(E)}{P(C)} \\
&= \arg\max_E \underbrace{P(C|E)}_{\text{translation model}}\underbrace{P(E)}_{\text{language model}}
\end{aligned}
$$

Typical Translation Models

- Word-based
  - Translating word to word
- Phrase-based
  - Translating based on phrase
- Syntax-based
  - Translating based on syntactic structure

Word-based Model: IBM Model One (Brown et al., 1993)

C: <NULL> 这 房子 很 小

$P(the|这)$

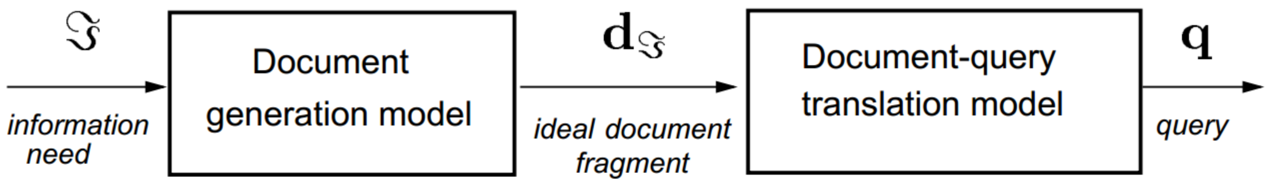
E: the house is very small

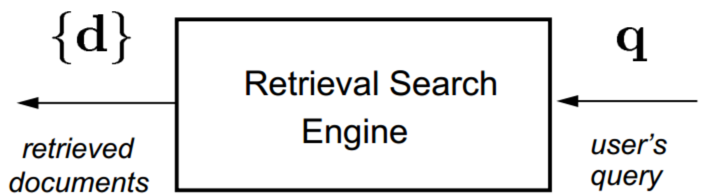
1 2 3 4 5

- Generating target sentence
  - Choose the length of target language I, according to $P(I|C)$
  - For each position, $i(i = 1, 2, \ldots, I)$
    - Choose position j in source sentence C according to $P(j|C)$
    - Generate target word e_i according to $P(e_j|c_i)$

$$P(E|C) = \frac{\epsilon}{(J+1)^I} \prod_{i=1}^{I} \sum_{j=1}^{J} P(e_i|c_j)$$

Model of Query Generation and Retrieval





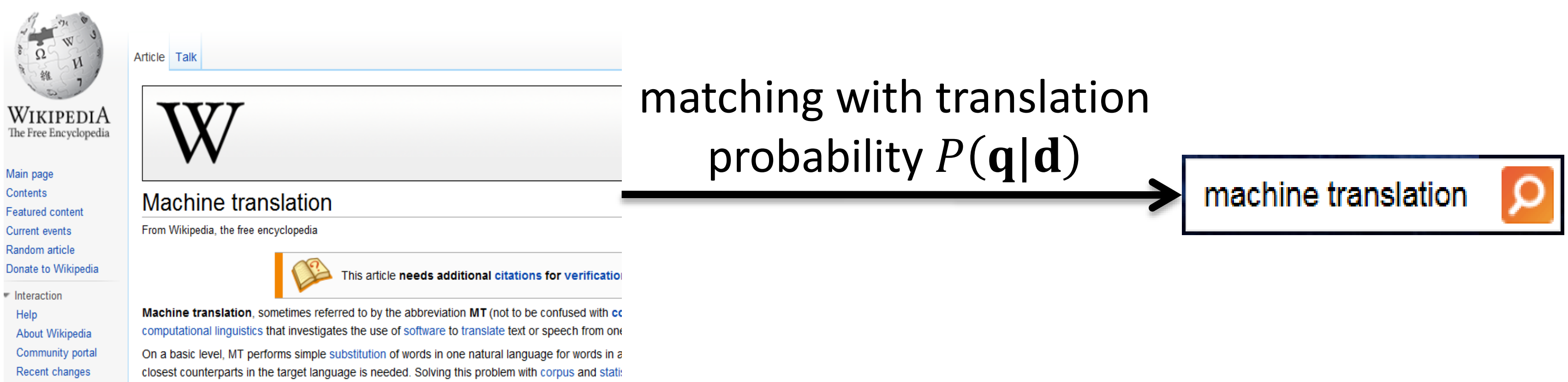
- Task of retrieval: find the a posteriori most likely documents given query

$$P(\mathbf{d}|\mathbf{q},\mathcal{U}) = \frac{P(\mathbf{q}|\mathbf{d},\mathcal{U})\cdot P(\mathbf{d}|\mathcal{U})}{P(\mathbf{q}|\mathcal{U})}$$

query dependent

query independent

Matching with Translation Model



- Translating document **d** to query **q**
- Given query **q** and document **d**, translation probability is viewed as matching score between **q** and **d**

Addressing Term Mismatch with Translation Model

- Translation probability $P(q|w)$ represents matching degree between words in query and document

q	$P(q\|w)$	q	$P(q\|w)$
titanic	0.56218	Vista	0.80575
ship	0.01383	Windows	0.05344
movie	0.01222	Download	0.00728
pictures	0.01211	ultimate	0.00571
sink	0.00697	xp	0.00355
facts	0.00689	microsoft	0.00342
photos	0.00533	bit	0.00286
rose	0.00447	compatible	0.00270
people	0.00441	premium	0.00244
survivors	0.00369	free	0.00211

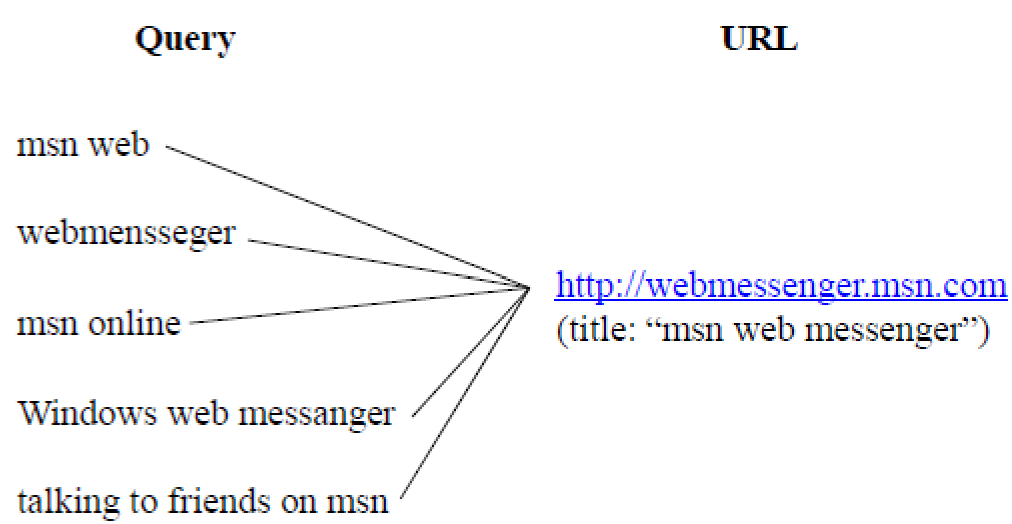
w = titanic w = vista

Issues Need to be Addressed

- Self-translation probability $P(w|w)$
  - Both source language and target language are in the same language
  - Too large: decrease effect of using translation
  - Too small: direct matching less effective and hurt the performance of matching

Issues Need to be Addressed

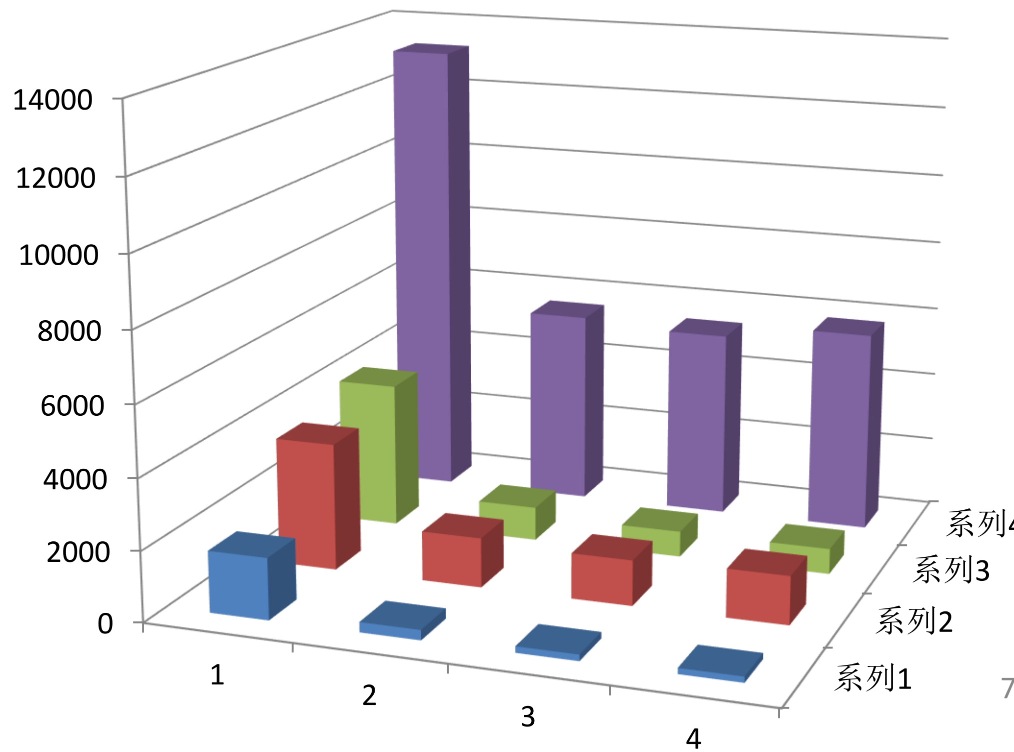
- Training data
  - Synthetic data (Berger & Lafferty, ’99)
  - Document collection (Karimzadehgan & Zhai, ’10)
  - Title-body pairs of documents (Jin et al., ’02)
  - Query-title pairs in click-through data (Gao et al., ’10)



Issues Need to be Addressed

- Document fields
  - Use of title is better than body (Huang et al., ‘10)
  - Titles and queries have similar languages
  - Bodies and queries have very different languages

$$Perplexity(\tilde{P}, Q) = 2^{H(\tilde{P},Q)}$$
$$= 2^{-\sum_s \tilde{p}_s \log q_s}$$



Methods for Matching with Translation

- Translating document to query
  - Word-based model (Berger & Lafferty, ’99; Gao et al., ‘10)
  - Phrase-based model (Gao et al., ’10)
  - Syntax-based model (Park et al., ’11)
  - Topic-based model (Gao et al., ’11)
  - Learning translation probabilities from documents (Karimzadehgan & Zhai, ’10)
- Translating document model to query model
  - Translated query language model (Jin et al., ’02)

Methods of Matching with Translation

- Basic model (Berger & Lafferty, '99)

$$P(q|d) = \frac{P(m|d)}{(n+1)^m} \prod_{j=1}^{m} \sum_{i=0}^{n} P(q_j|d_i)$$

$$= P(m|d) \prod_{j=1}^{m} \left(\frac{n}{n+1} P(q_j|d) + \frac{1}{n+1} P(q_j|\langle null \rangle) \right)$$

Word q_j being translated from document d.
$P(q_j|d) = \sum_{w \in d} P(q_j|w) Q(w|d)$
$P(q_j|w)$: probability of w being translated to q_j
$Q(w|d)$: un-smoothed document language model

Smoothing to avoid zero probability

- Adding self-translation (Gao et al., '10)

$$P'(q_j|d) = \beta Q(q_j|d) + (1-\beta) \sum_{w \in d} P(q_j|w) Q(w|d)$$

Un-smoothed document language model

Performances of Word-based Translation Model in Search

	NDCG@1	NDCG@3	NDCG@10
BM25 (baseline)	0.3181	0.3413	0.4045
WTM (without self-translation)	0.3210	0.3512	0.4211
WTM (with self-translation)	**0.3310**	**0.3566**	**0.4232**

- Evaluation based on 12071 real queries
- WTM can outperform baseline of BM25
- WTM can be further improved by self-translation

Examples of Translation Probabilities

q	$P(q\|w)$	q	$P(q\|w)$
titanic	0.56218	Vista	0.80575
ship	0.01383	Windows	0.05344
movie	0.01222	Download	0.00728
pictures	0.01211	ultimate	0.00571
sink	0.00697	xp	0.00355
facts	0.00689	microsoft	0.00342
photos	0.00533	bit	0.00286
rose	0.00447	compatible	0.00270
people	0.00441	premium	0.00244
survivors	0.00369	free	0.00211

w = titanic — w = vista

References

- Adam Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal. Bridging the Lexical Chasm: Statistical Approaches to Answer-Finding. In SIGIR 2000.
- Adam Berger and John Lafferty. Information Retrieval as Statistical Translation. In SIGIR 1999.
- Jianfeng Gao, Xiaodong He, and JianYun Nie. Click-through-based Translation Models for Web Search: from Word Models to Phrase Models. In CIKM 2010.
- Jianfeng Gao, Kristina Toutanova, and Wen-tau Yih. Clickthrough-based latent semantic models for web search. In SIGIR 2011.
- Jianfeng Gao : Statistical Translation and Web Search Ranking. http://research.microsoft.com/en-us/um/people/jfgao/paper/SMT4IR.res.pptx
- Jianfeng Gao and Jian-Yun Nie. 2012. Towards concept-based translation models using search logs for query expansion. In CIKM 2012.
- Jianfeng Gao, Shasha Xie, Xiaodong He and Alnur Ali. 2012. Learning lexicon models from search logs for query expansion. In EMNLP 2012.
- Dustin Hillard, Stefan Schroedl, and Eren Manavoglu, Hema Raghavan, and Chris Leggetter. Imrpoved Ad Relevance in Sponsored Search. In WSDM 2010.
- Jian Huang, Jianfeng Gao, Jiangbo Miao, Xiaolong Li, Kuansan Wang, Fritz Behr, and C. Lee Giles. Exploring web scale language models for search query processing. In WWW 2010.
- Ea-Ee Jan, Shih-Hsiang Lin, and Berlin Chen. Translation Retrieval Model for Cross Lingual Information Retrieval. In AIRS 2010.

References

- Rong Jin, Alex G. Hauptmann, and Chengxiang Zhai. Title Language Model for Information Retrieval. In SIGIR 2002.
- Maryan Karimzadehgan and Chengxiang Zhai. Estimation of Statistical Translation Models based on Mutual Information for Ad Hoc Information Retrieval. In SIGIR 2010.
- David Mimno , Hanna M. Wallach , Jason Naradowsky , David A. Smith, Andrew McCallum. Polylingual topic models. In EMNLP 2009.
- Seung-Hoon Na and Hwee Tou Ng. Enriching Document Representation via Translation for Improved Monolingual Information Retrieval. In SIGIR 2011.
- Jae-Hyun Park, W. Bruce Croft, and David A. Smith. Qusi-Synchronous Dependence Model for Information Retrieval. In CIKM 2011.
- Stefan Riezler and Yi Liu. Query Rewritting Using Monolingual Statistical Machine Translation. In ACL 2010.
- Dolf Trieschnigg, Djoerd Hiemstra, Franciska de Jong, and Wessel Kraaij. A cross-lingual Framework for Monolingual Biomedical Information Retrieval. In CIKM 2010.
- Elisabeth Wolf, Delphine Bernhard, and Iryan Gurevych. Combining Probabilistic and Translation-based Models for Information Retrieval based on Word Sense Annotations. In CLEF Workshop 2009.

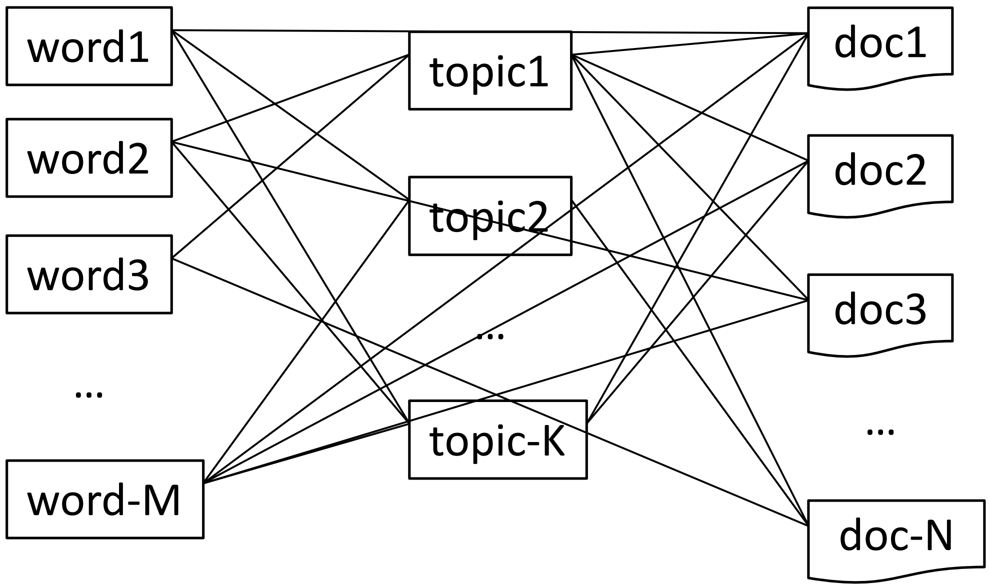
Outline of Tutorial

- Semantic Matching between Query and Document
- Approaches to Semantic Matching
  1. Matching by Query Reformulation
  2. Matching with Term Dependency Model
  3. Matching with Translation Model
  4. Matching with Topic Model
  5. Matching with Latent Space Model
- Summary

Outline

- Topic Models
- Methods of Matching with Topic Model

Topic Modeling

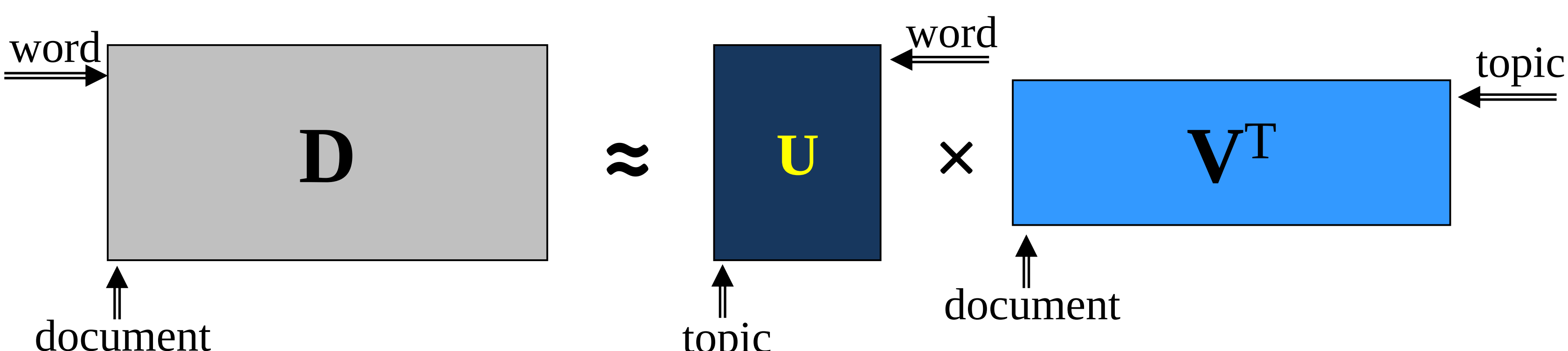


- Input
  - Document collection
- Processing
  - Discover latent topics in document collection
- Output
  - Latent topics in document collection
  - Topic representations of documents

Two Approaches

- Probabilistic approach

- Non-probabilistic approach



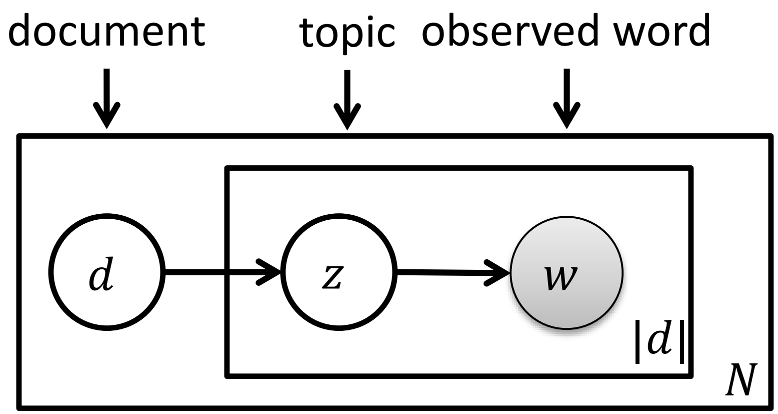
Topic Modeling: Two Approaches (cont')

- Probabilistic Topic Models
  - Model: probabilistic model (graphical model)
  - Learning: maximum likelihood estimation
  - Methods: PLSI, LDA
- Non-probabilistic Topic Models
  - Model: vector space model
  - Learning: matrix factorization
  - Methods: LSI, NMF, RLSI
- Non-probabilistic models can be reformulated as probabilistic models

Probabilistic Topic Model

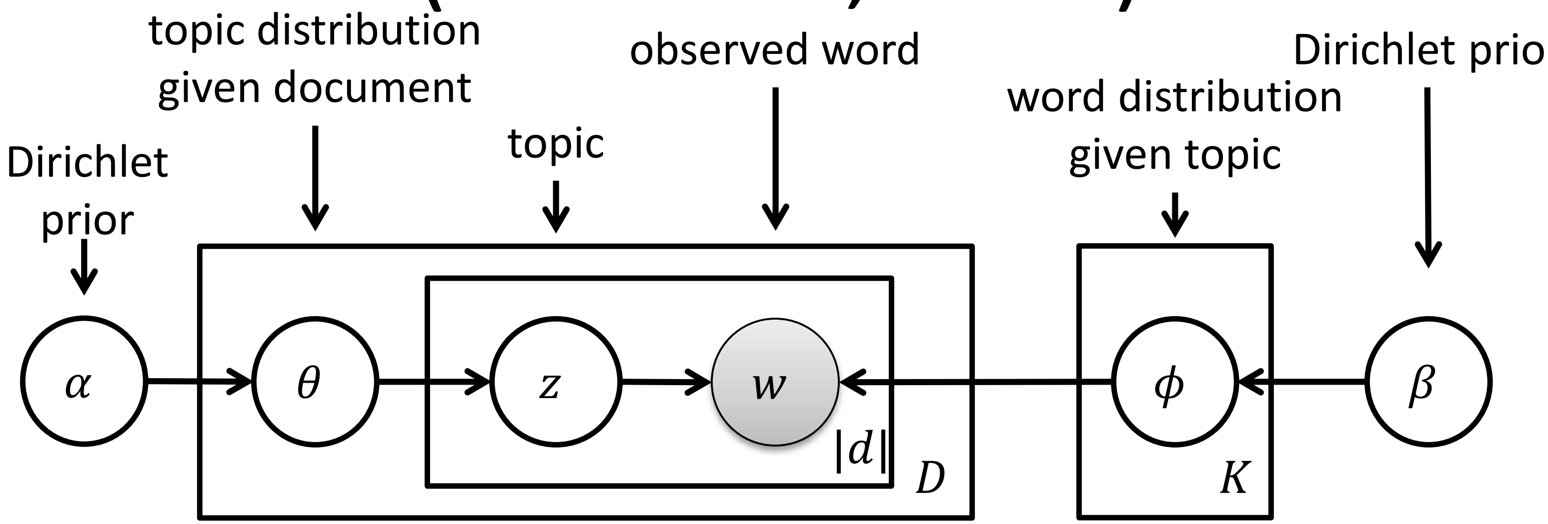
- Topic: probability distribution over words
- Document: probability distribution over topics
- Graphical model
  - Word, topic, document, and topic distribution are represented as nodes
  - Probabilistic dependencies are represented as directed edges
  - Generation process
- Interpretation: soft clustering

Probabilistic Latent Semantic Indexing (Hofmann 1999)



1. select a document d from the collection with probability $P(d)$
2. for each document d in the collection
   (a) select a latent topic z with probability $P(z|d)$
   (b) generate a word w with probability $P(w|z)$

Latent Dirichlet Allocation (Blei et al., 2003)



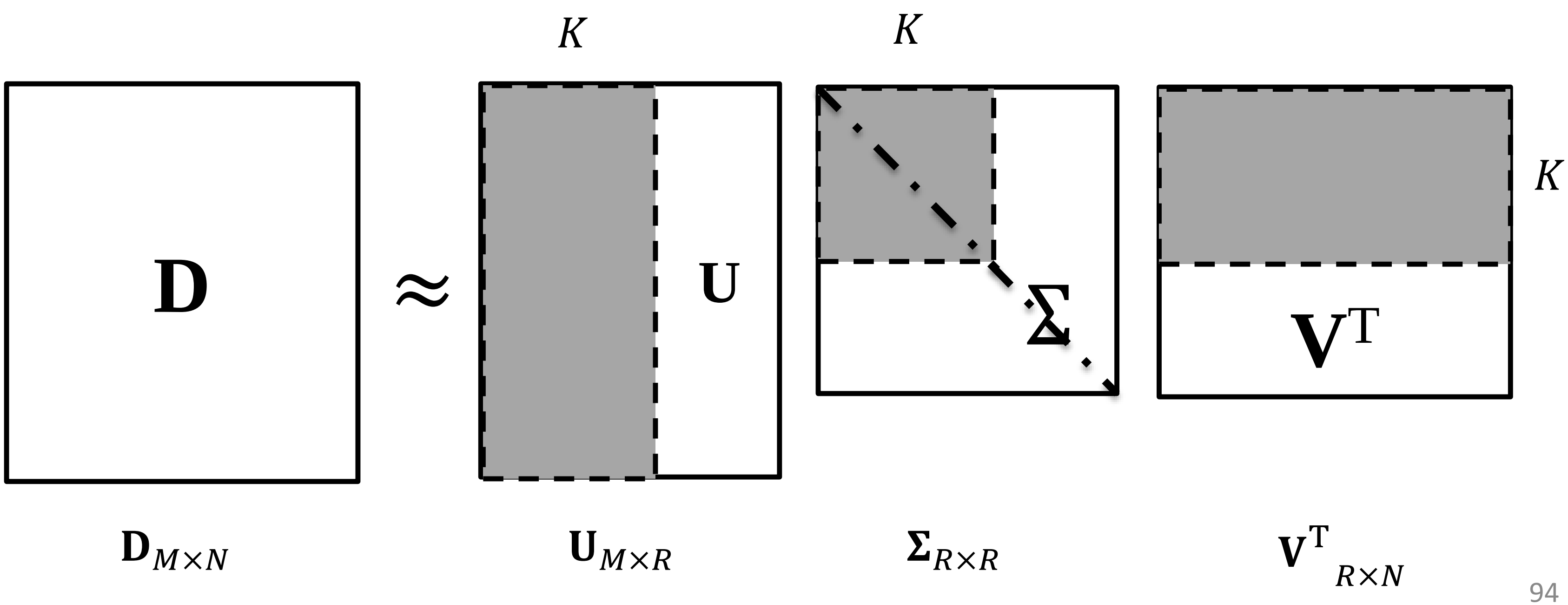
1. for each topic $k = 1, \cdots, K$
   (a) draw word distribution ϕ_k according to $\phi_k|\beta \sim Dir(\beta)$
2. for each document d in the collection
   (a) draw topic distribution θ according to $\theta|\alpha \sim Dir(\alpha)$
   (b) for each word w in the document d
      i. draw a topic z according to $z|\theta \sim Mult(\theta)$
      ii. draw a word w according to $w|z, \phi_{1:K} \sim Mult(\phi_z)$

Non-probabilistic Topic Model

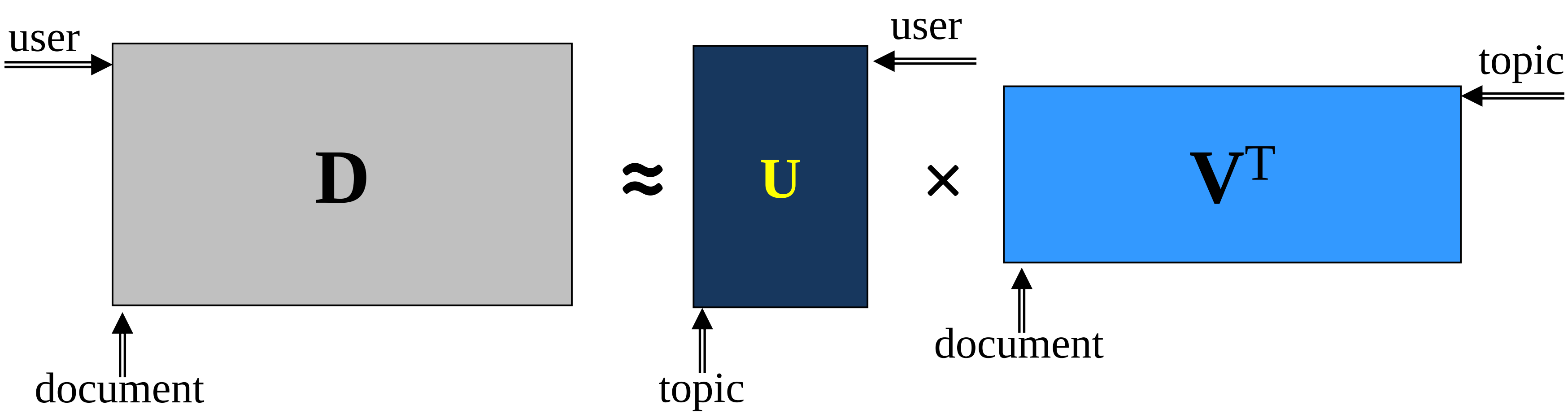
- Document: vector of words
- Topic: vector of words
- Document representation: combination of topic vectors
- Matrix factorization
- Interpretation: projection to topic space

Latent Semantic Indexing (Deerwester et al., 1990)

- Representing document collection with co-occurrence matrix (TF or TFIDF)
- Performing Singular Value Decomposition (SVD) and producing k-dimensional topic space



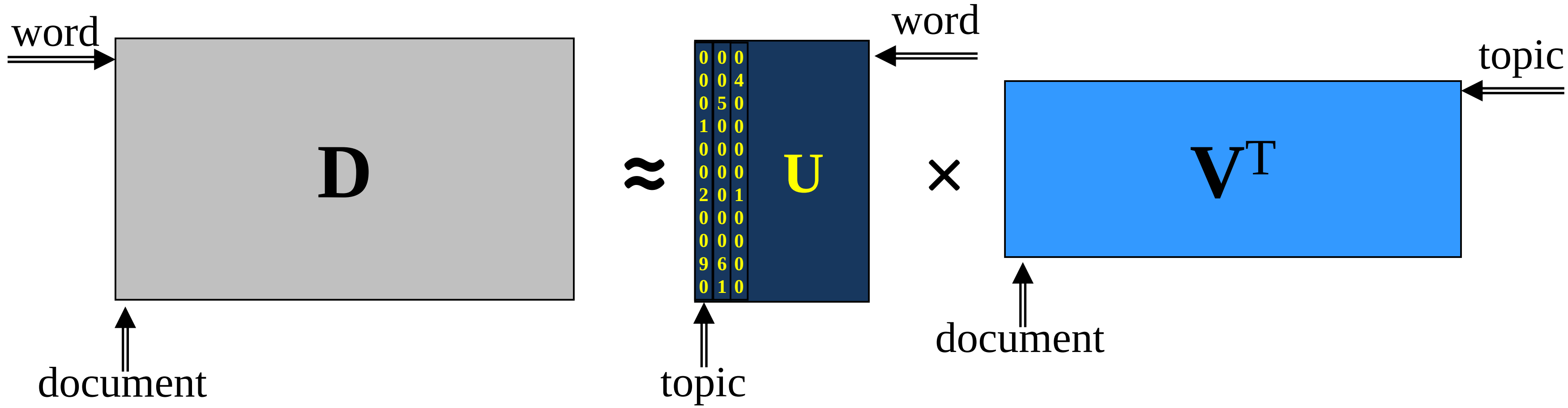
Nonnegative Matrix Factorization (Lee and Seung, 2001)



- **U** and **V** are nonnegative

$$\min_{\mathbf{U},\mathbf{V}} \left\| \mathbf{D} - \mathbf{U}\mathbf{V}^{\mathrm{T}} \right\|_F$$
$$s.t.\ u_{ij} \geq 0; v_{ij} \geq 0$$

Regularized Latent Semantic Indexing (Wang et al., 2011)



- Topics are sparse

word representation of doc n

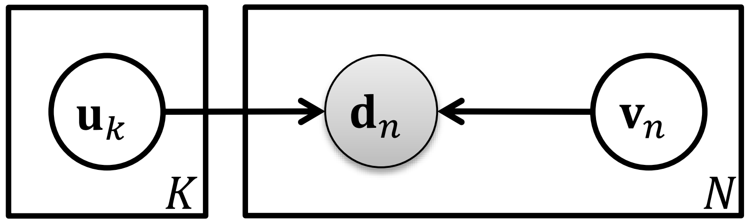
topic matrix

topic representation of doc n

$$\min_{\mathbf{U},\mathbf{V}} \sum_{n=1}^{N} \|\mathbf{d}_n - \mathbf{U}\mathbf{v}_n\|_2^2 + \lambda_1 \sum_{k=1}^{K} \|\mathbf{u}_k\|_1 + \lambda_2 \sum_{n=1}^{N} \|\mathbf{v}_n\|_2^2$$

topics are sparse

Probabilistic Interpretation of Nonprobabilistic Models (RLSI)



$$\min_{\mathbf{U},\mathbf{V}} \sum_{n=1}^{N} \|\mathbf{d}_n - \mathbf{U}\mathbf{v}_n\|_2^2 + \lambda_1 \sum_{k=1}^{K} \|\mathbf{u}_k\|_1 + \lambda_2 \sum_{n=1}^{N} \|\mathbf{v}_n\|_2^2$$

- Document generated according to Gaussian distribution

$$P(\mathbf{d}_n|\mathbf{U},\mathbf{v}_n) \propto \exp(-\|\mathbf{d}_n - \mathbf{U}\mathbf{v}_n\|_2^2)$$

- Laplacian prior

$$P(\mathbf{u}_k) \propto \exp(-\lambda_1\|\mathbf{u}_k\|_1)$$

- Gaussian prior

$$P(\mathbf{v}_n) \propto \exp(-\lambda_2\|\mathbf{v}_n\|_2^2)$$

Deal with Term Mismatch with Topic Model

- Topics of query and document are identified
- Match query and document through topics, although query and document do not share terms
- Linear combination with term model

$$s(q,d) = \alpha s_{topic}(q,d) + (1-\alpha) s_{term}(q,d)$$

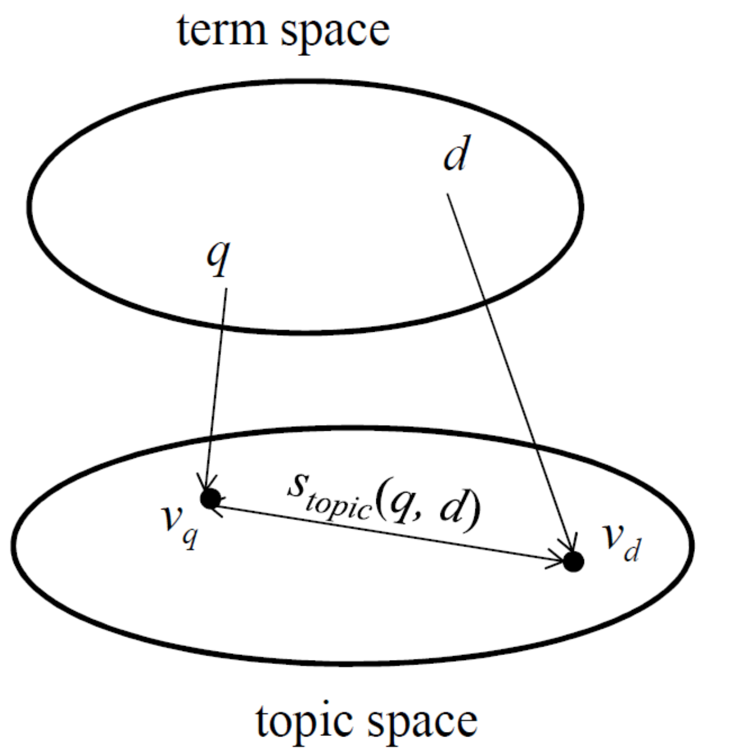
Topic1	Topic2	Topic3	Topic4	Topic5	Topic6	Topic7	Topic8	Topic9	Topic10
OPEC	Africa	contra	school	Noriega	firefight	plane	Saturday	Iran	senate
oil	South	Sandinista	student	Panama	ACR	crash	coastal	Iranian	Reagan
cent	African	rebel	teacher	Panamanian	forest	flight	estimate	Iraq	billion
barrel	Angola	Nicaragua	education	Delval	park	air	western	hostage	budget
price	apartheid	Nicaraguan	college	canal	blaze	airline	Minsch	Iraqi	Trade

Methods of Matching Using Topic Model

- Topic matching
  - Probabilistic model: PLSI (Hofmann ’99), LDA (Blei et al., ’03)
  - Non-probabilistic model: LSI (Deerwester et al., ’88), NMF (Lee & Seung ’00), RLSI (Wang et al., ’11), GMF (Wang et al., ’12)
- Smoothing
  - Clustering-based (Kurland & Lee ’04, Diaz ’05)
  - LDA-based (Wei & Croft ’06)
  - PLSI-based (Yi & Allan ’09)

Topic Level Matching

- Representing query and document as topic vectors (or topic distributions)
- Calculating matching score in topic space



Topic Level Matching (cont')

- In RLSI, query and document representation
  - $\mathbf{q} \to v_q = \left(\mathbf{U}^{\mathrm{T}}\mathbf{U} + \lambda_2 \mathbf{I}\right)^{-1} q$
  - $\mathbf{d} \to v_d = \left(\mathbf{U}^{\mathrm{T}}\mathbf{U} + \lambda_2 \mathbf{I}\right)^{-1} d$
- Topic level matching
  - Cosine similarity

$$s_{topic}(q, d) = \frac{\langle v_q, v_d \rangle}{\|v_q\|_2 \|v_d\|_2}$$

  - Symmetric KL-divergence

$$s_{topic}(q, d) \quad = \quad 1 - \frac{1}{2}\left(\mathrm{KL}(v_q \| v_d) + \mathrm{KL}(v_d \| v_q)\right)$$

Experimental Results

	MAP	NDCG@1	NDCG@3	NDCG@5	NDCG@10
BM25	0.3918	0.4400	0.4268	0.4298	0.4257
BM25+LSI	0.3952	0.4720	0.4410	0.4360	0.4365
BM25+NMF	0.3985*	0.4600	0.4445*	0.4408*	0.4347*
BM25+PLSI	0.3928	0.4680	0.4383	0.4351	0.4291
BM25+LDA	0.3952	0.4760*	0.4478*	0.4332	0.4292
BM25+RLSI	**0.3998***	**0.4800***	**0.4461***	**0.4498***	**0.4420***

- Topic models can improve the baseline of BM25
- LDA, NMF, and RLSI perform slightly better than the others

References

- Paul N. Bennett, Krysta Svore, and Susan T. Dumais. Classification Enhanced Ranking. In Proc. of WWW 2010.
- David Blei, Andrew Ng, Michael Jordan, John Lafferty. Latent Dirichlet allocation. JMLR, 2003.
- Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Richard Harshman. Indexing by Latent Semantic Analysis, JASIST, 1990.
- Fernando Diaz. Regularizing ad hoc retrieval scores. In Proc. of CIKM 2005.
- Martin Franz and Jeffery S McCarley. Information retrieval with non-negative matrix factorization. IBM Patent. 2001.
- Thomas Hofmann, Probabilistic Latent Semantic Indexing. In Proc. of SIGIR 1999.
- Oren Kurland and Lillian Lee. Corpus Structure, Language Models, and Ad Hoc Information Retrieval. In Proc. of SIGIR 2004.
- Daniel D. Lee and H. Sebastian Seung. Algorithms for Non-negative Matrix Factorization. In Proc. of NIPS 2000.
- Michael L. Littman, Susan T. Dumais, and Thomas K. Landauer. Automatic Cross-Language Information Retrieval using Latent Semantic Indexing. Cross-Language Information Retrieval, 1996.

References

- Xiaoyong Liu and W. Bruce Croft. Cluster-based Retrieval using Language models. In Proc. of SIGIR 2004.
- David Mimno, Hanna Wallach, Jason Naradowsky, David Smith, and Andrew McCallum. Ploylingual Topic models. In Proc. of EMNLP 2009
- Quan Wang, Jun Xu, Hang Li, and Nick Craswell. Regularized Latent Semantic Indexing. In Proc. of SIGIR 2011.
- Xing Wei and W. Bruce Croft. LDA-based Document Models for Ad-hoc Retrieval. In Proc. of SIGIR 2006.
- Jinxi Xu and W. Bruce Croft. Cluster-based Language Models for Distributed Retrieval. In Proc. of SIGIR 1999.
- Xing Yi and James Allan. A comparative study of utilizing topic models for Information Retrieval. In Proc. of ECIR 2009.

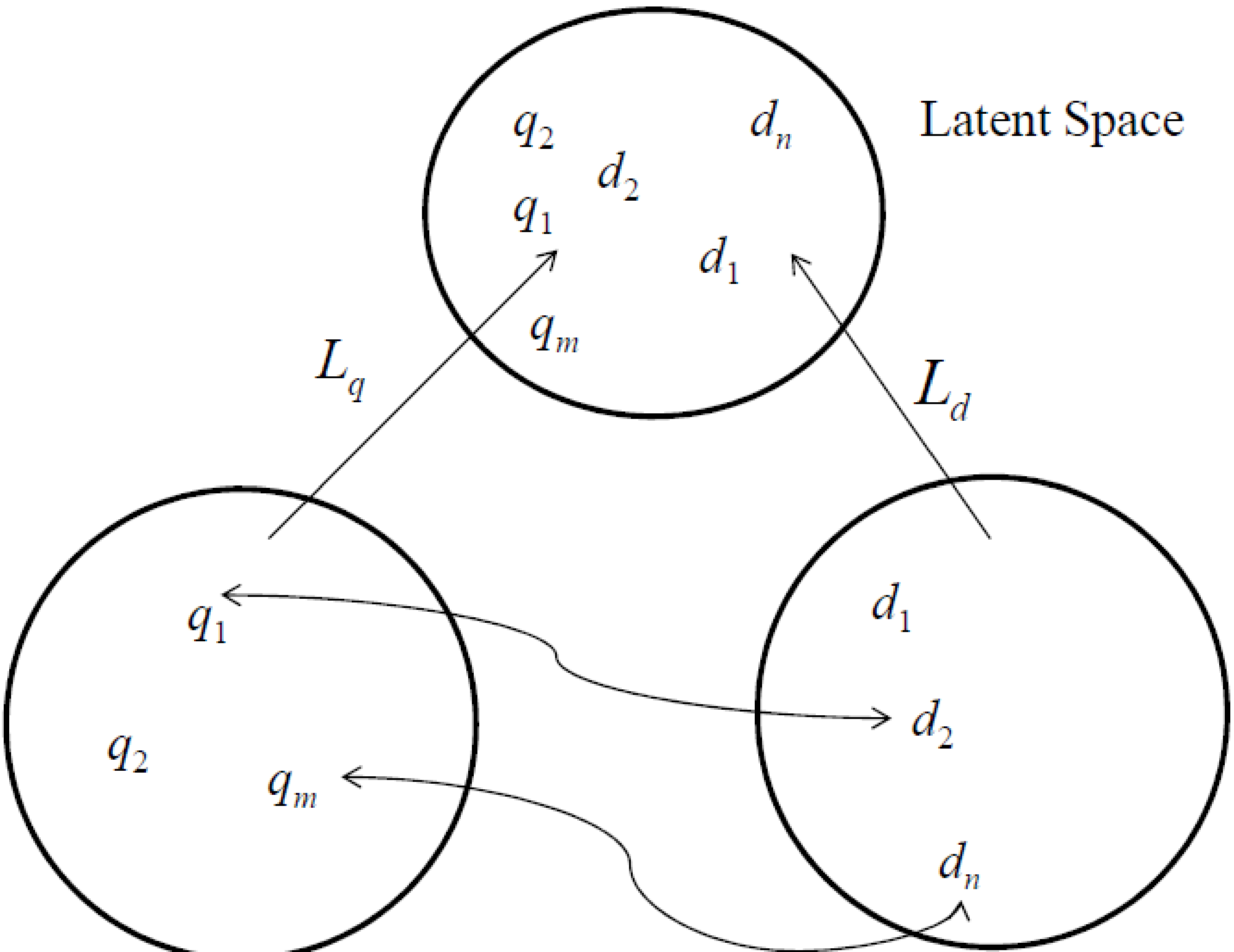
Outline of Tutorial

- Semantic Matching between Query and Document
- Approaches to Semantic Matching
  1. Matching by Query Reformulation
  2. Matching with Term Dependency Model
  3. Matching with Translation Model
  4. Matching with Topic Model
  5. Matching with Latent Space Model
- Summary

Matching in Latent Space

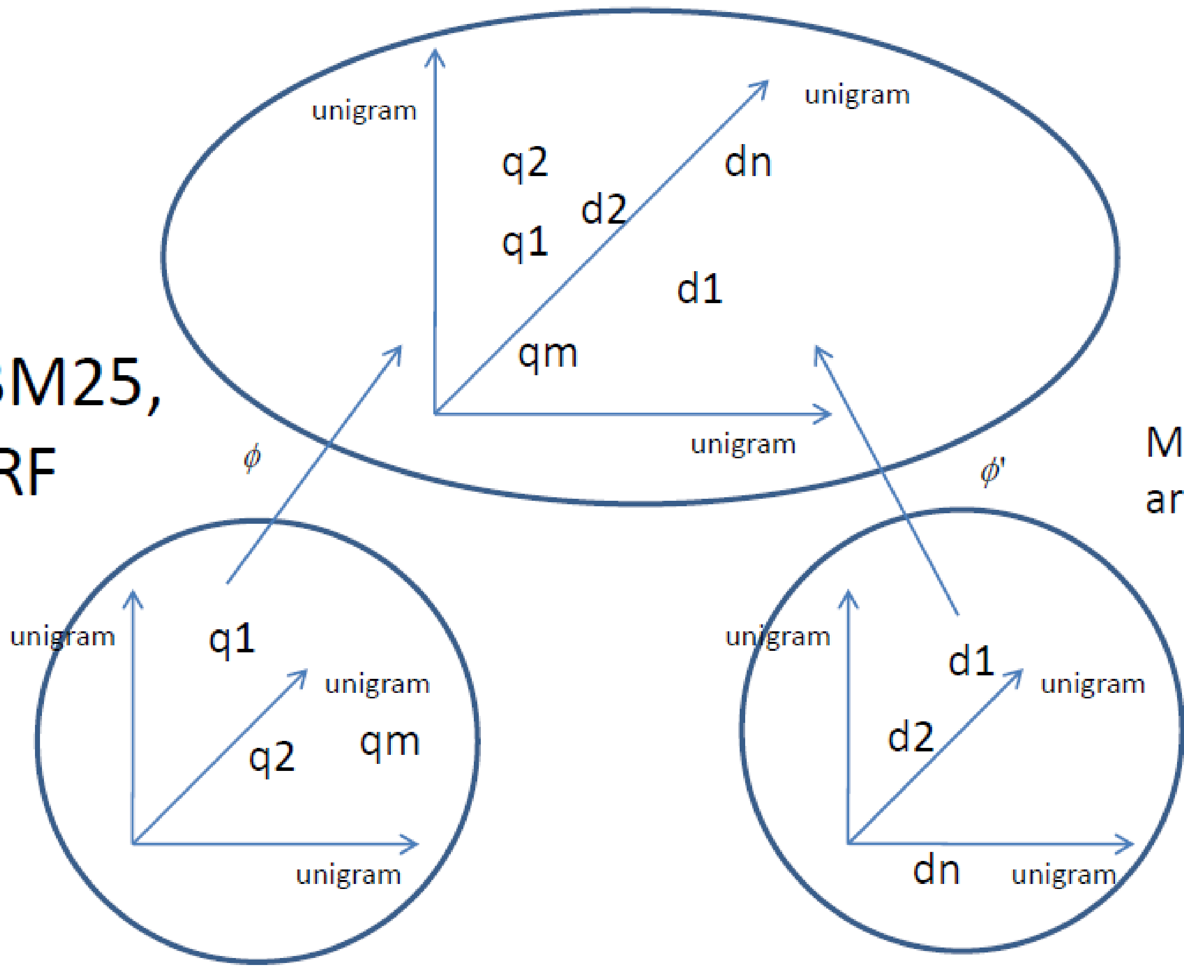
- Motivation
  - Matching between query and document in latent space
- Assumption
  - Queries have similarity
  - Documents have similarity
  - Click-through data represent “similarity” relations between queries and documents
- Approach
  - Projection to latent space
  - Regularization or constraints
- Results
  - Significantly enhance accuracy of query document matching

Matching in Latent Space



IR Models as Similarity Functions (Xu et al., 2010)

VSM, BM25, LM, MRF



New Space

Mapping functions are diagonal matrices

Query Space

Document Space

IR Models as Similarity Functions

- VSM

$$f_{\mathtt{VSM}}(q,d) = \langle \phi_{\mathtt{VSM}}(q), \phi'_{\mathtt{VSM}}(d) \rangle = \langle q, d \rangle.$$

- BM25

$$f_{\mathtt{BM25}}(q,d) = \langle \phi_{\mathtt{BM25}}(q), \phi'_{\mathtt{BM25}}(d) \rangle$$

$$\phi_{\mathtt{BM25}}(q)_x = \frac{(k_3+1)\cdot f(x,q)}{k_3 + f(x,q)}$$

$$\phi'_{\mathtt{BM25}}(d)_x = \mathtt{IDF}(x) \cdot \frac{(k_1+1)\cdot f(x,d)}{k_1\left(1-b+b\frac{f(d)}{avgf}\right) + f(x,d)}$$

Deal with Term Mismatch with Latent Space Model

- Matching in Latent Space can solve the problem by
  - Reducing dimensionality of latent space (from term level matching to semantic matching)
  - Correlating semantically similar terms (matrices are not diagonal)
  - Automatically learning mapping functions from data
- *Generalized and Learnable of IR models*

Partial Least Square (PLS)

- Input
  - Training data: $\{(q_i, d_i, c_i)\}_{1 \le i \le N}, q_i \in Q, d_i \in D,\ c_i \in \{+1, -1\}\ or\ c_i \in R$
- Output
  - Similarity function $f(q, d)$
- Assumption
  - Two linear and orthonormal transformations L_q and L_d
  - Dot product as similarity function $f(q, d) = \langle L_q \cdot q, L_d \cdot d \rangle$
- Optimization

$$\arg\max_{L_q, L_d} = \sum_{(q_i, d_i)} c_i f(q_i, d_i),$$

$$L_q L_q^T = I, \quad L_d L_d^T = I$$

Solution of Partial Least Square

- Non-convex optimization
- Can prove that global optimal solution exists
- Global optimal can be found by solving SVD
- SVD of matrix $M_S - M_D = U\Sigma V^T$

Regularized Mapping to Latent Space (Wu et al., '13)

- Input
  - Training data: $\{(q_i, d_i, c_i)\}_{1 \le i \le N}, q_i \in Q, d_i \in D,\ c_i \in \{+1, -1\}\ or\ c_i \in R$
- Output
  - Similarity function $f(q, d)$
- Assumption
  - ℓ_1 and ℓ_2 regularization on L_X and L_Y (sparse transformations)
  - Dot product as similarity function $f(q, d) = \langle L_q \cdot q, L_d \cdot d \rangle$
- Optimization

$$\arg\max_{L_q, L_d} = \sum_{(q_i, d_i)} c_i f(q_i, d_i),$$

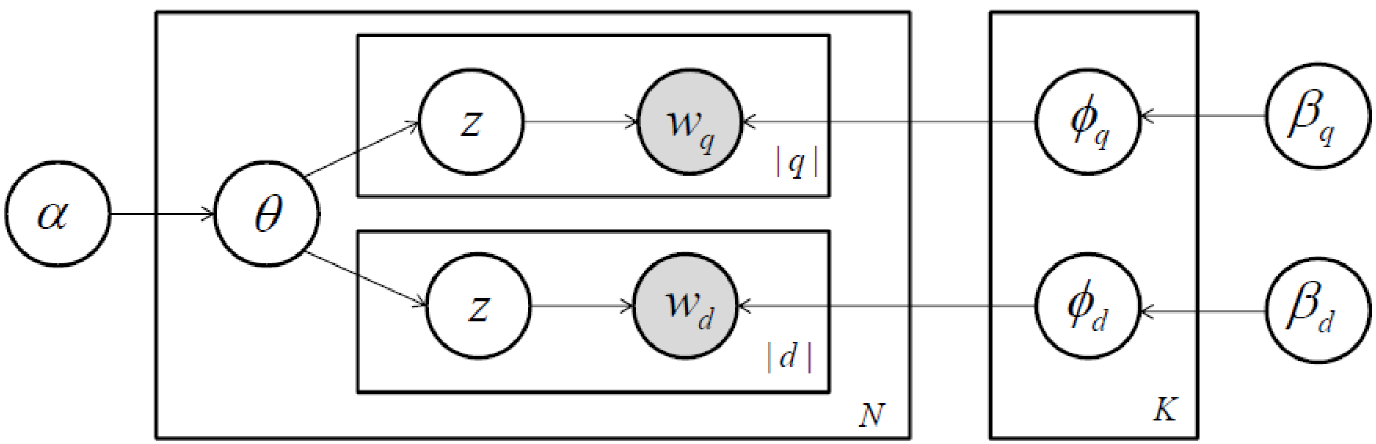
$$|l_q| \le \theta_q, \quad |l_d| \le \theta_d, \quad \|l_q\| \le \tau_q, \quad \|l_d\| \le \tau_d$$

Solution of Regularized Mapping to Latent Space

- Coordinate Descent
- Repeat
  - Fix L_X, update L_Y
  - Fix L_Y, update L_X
- Update can be parallelized by rows

Bilingual Topic Model (Gao et al., '11)

- A natural extension of LDA for generating pairs of documents
- Each query document pair is generated from the same distribution of topics
- EM algorithm can be employed to estimate the parameters



$$P(\mathbf{q}|\mathbf{d}) = \prod_{q \in \mathbf{q}} P_{bltm}(q|\mathbf{d}) = \prod_{q \in \mathbf{q}} \sum_{z} P\left(q|\phi_z^{\mathbf{q}}\right) P\left(z|\theta^{\mathbf{d}}\right)$$

Comparison

	PLS	**RMLS**	**BLTM**
Assumption	Orthogonal	ℓ_1 and ℓ_2 regularization	Topic Modeling
Optimization Method	Singular Value Decomposition	Coordinate Descent	EM
Optimality	Global optimum	Local optimum	Local optimum
Efficiency	Low	High	Low
Scalability	Low	High	Low

Experimental Results

Table 7.1: Performances of latent space models in search.

	NDCG@1	NDCG@3	NDCG@5
BM25 (baseline)	0.637	0.690	0.690
SSI	0.538	0.621	0.629
SVDFeature	0.663	0.720	0.727
BLTM	0.657	0.702	0.701
PLS	0.676	0.728	**0.736**
RMLS	**0.686**	**0.732**	0.729

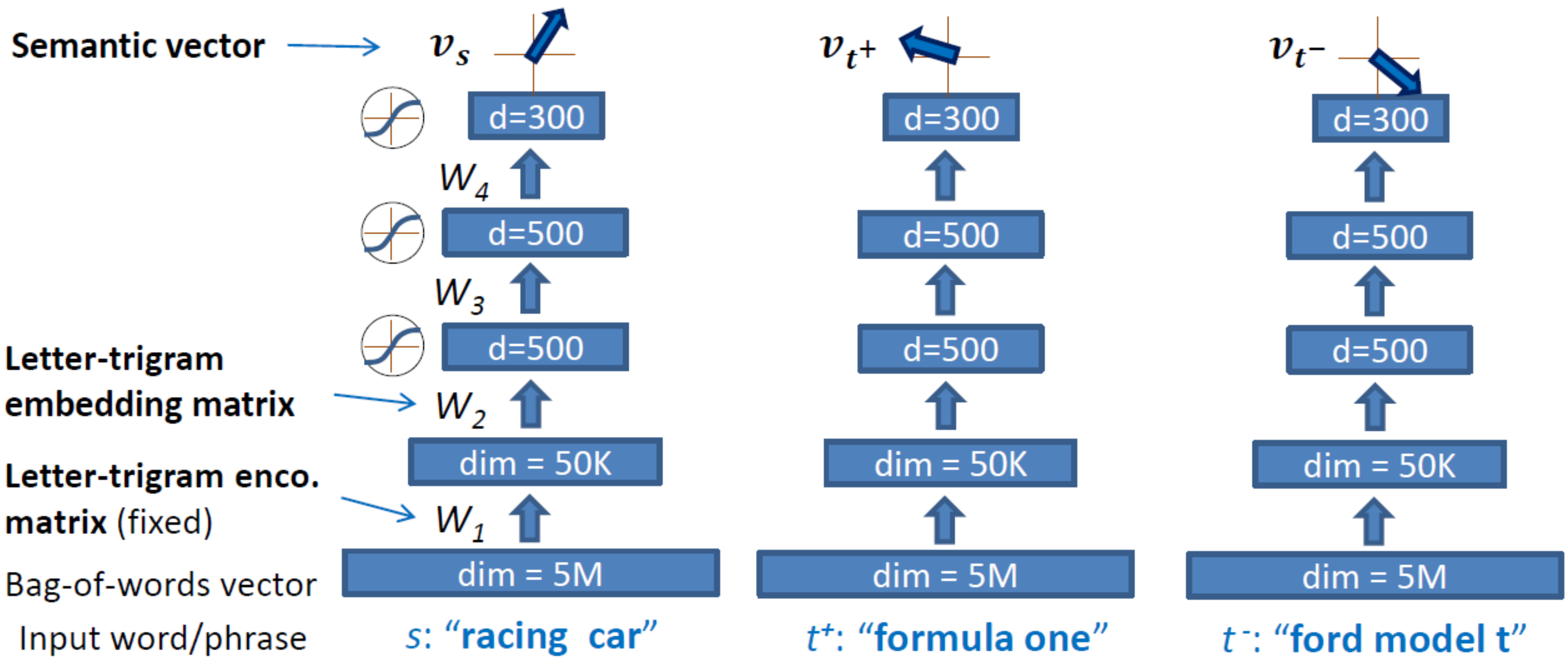
- 94,022 queries, 111,631 documents, and click through data;
- RMLS and PLS work better than BM25, SSI, SVDFeature, and BLTM
- RMLS works equally well as PLS, with higher learning efficiency and scalability

Learning Semantic Embedding using the DSSM

[Huang, He, Gao, Deng, Acero, Heck, 2013]

Initialization:

Neural networks are initialized with random weights

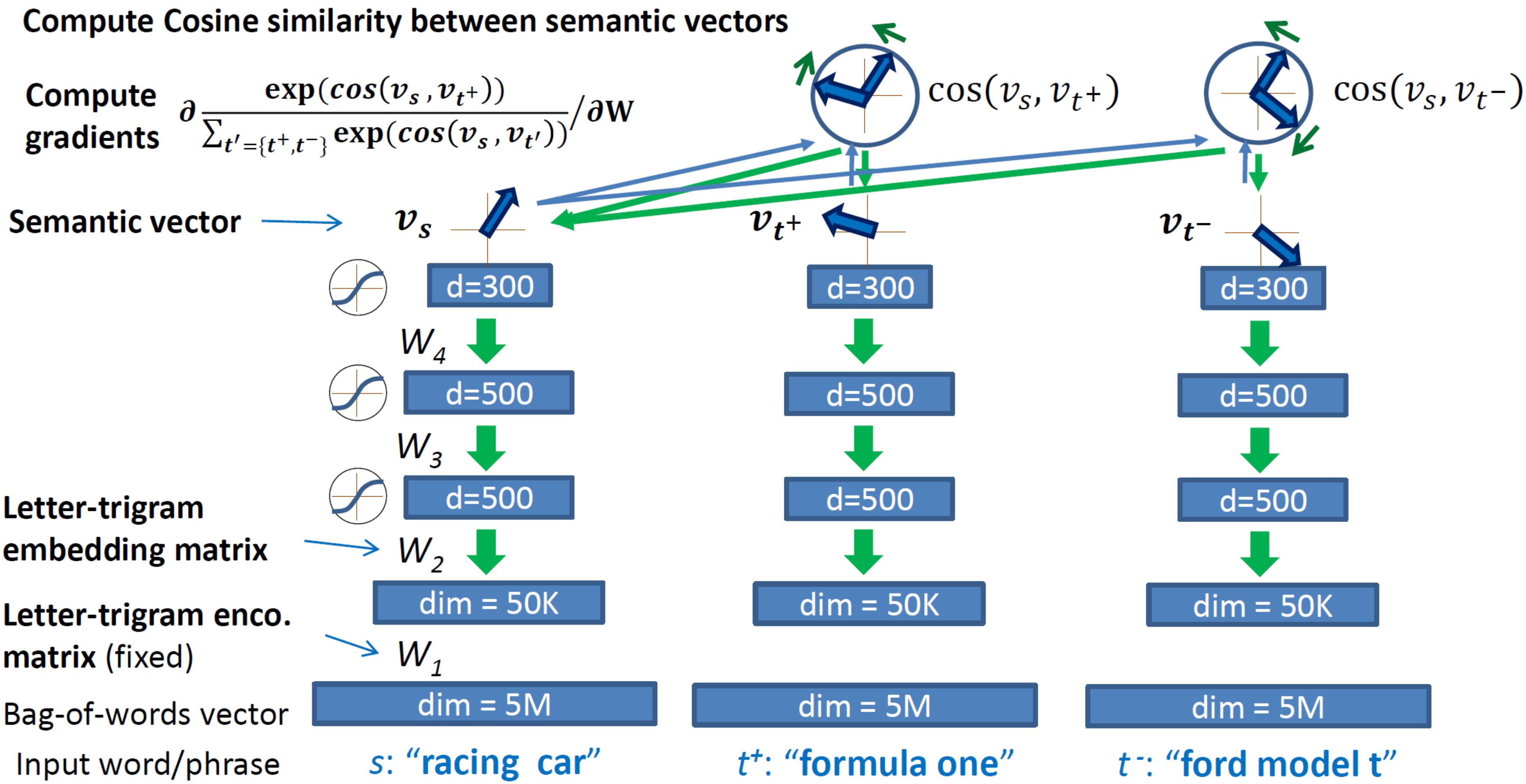


From Jianfeng Gao, CIKM 2014

Learning Semantic Embedding using the DSSM

Training (Back Propagation):

[Huang, He, Gao, Deng, Acero, Heck, 2013]

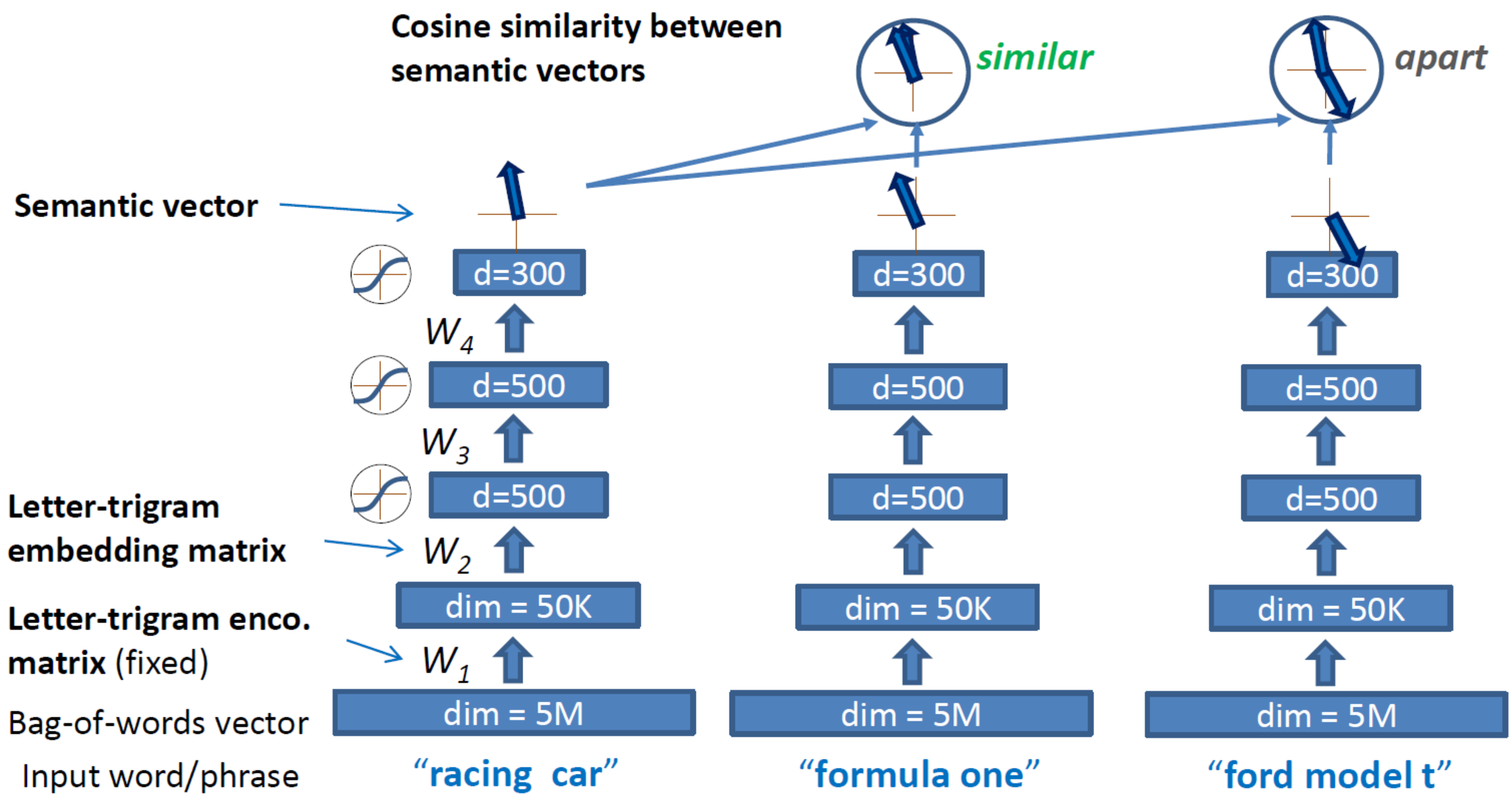


From Jianfeng Gao, CIKM 2014

Learning Semantic Embedding using the DSSM

[Huang, He, Gao, Deng, Acero, Heck, 2013]

After training converged:



From Jianfeng Gao, CIKM 2014

Experimental Results

Table 7.2: Performances of latent space models in search.

	NDCG@1	NDCG@3	NDCG@10
BM25 (baseline)	0.308	0.373	0.455
WTM	0.332	0.400	0.478
LSI	0.298	0.372	0.455
PLSI	0.295	0.371	0.456
BLTM	0.337	0.403	0.480
DSSM (linear)	0.357	0.422	0.495
DSSM (non-linear)	**0.362**	**0.425**	**0.498**

- Experiments conducted with 16510 queries, and each query on average associated with 15 webpages
- DSSM outperformed all baselines
- DSSM (non-linear) is the best

References

- Bing Bai, Jason Weston, David Grangier, Ronan Collobert, Kunihiko Sadamasa, Yanjun Qi, Olivier Chapelle, and Kilian Weinberger. Supervised semantic indexing. In CIKM ’09, pages 187–196, 2009.
- David R. Hardoon, Sandor R. Szedmak, and John R. Shawe-taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Comput.*, 16(12):2639–2664, 2004.
- Jun Xu, Hang Li, and Chaoliang Zhong. Relevance ranking using kernels. In volume 6458 of Lecture Notes in Computer Science, pages 1–12, 2010.
- Roman Rosipal and Nicole Krämer. Overview and recent advances in partial least squares. In SLSFS’05, pages 34–51, 2006.
- Wei Wu, Zhengdong Lu, and Hang Li. Learning bilinear model for matching queries and documents. J. Mach. Learn. Res., 14(1):2519–2548, 2013.
- Jianfeng Gao, Kristina Toutanova, and Wen-tau Yih. Clickthrough-based latent semantic models for web search. In SIGIR ’11, pages 675–684, 2011.
- Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In CIKM ’13, pages 2333–2338, 2013.

Outline of Tutorial

- Semantic Matching between Query and Document
- Approaches to Semantic Matching
  1. Matching by Query Reformulation
  2. Matching with Term Dependency Model
  3. Matching with Translation Model
  4. Matching with Topic Model
  5. Matching with Latent Space Model
- Summary

Summary of Tutorial

- Query document matching is one of the biggest challenge in search
- Machine learning for matching between query and document is making progress
- Matching at form, phrase, sense, topic, and structure aspects
- General problem: learning to match

Approaches

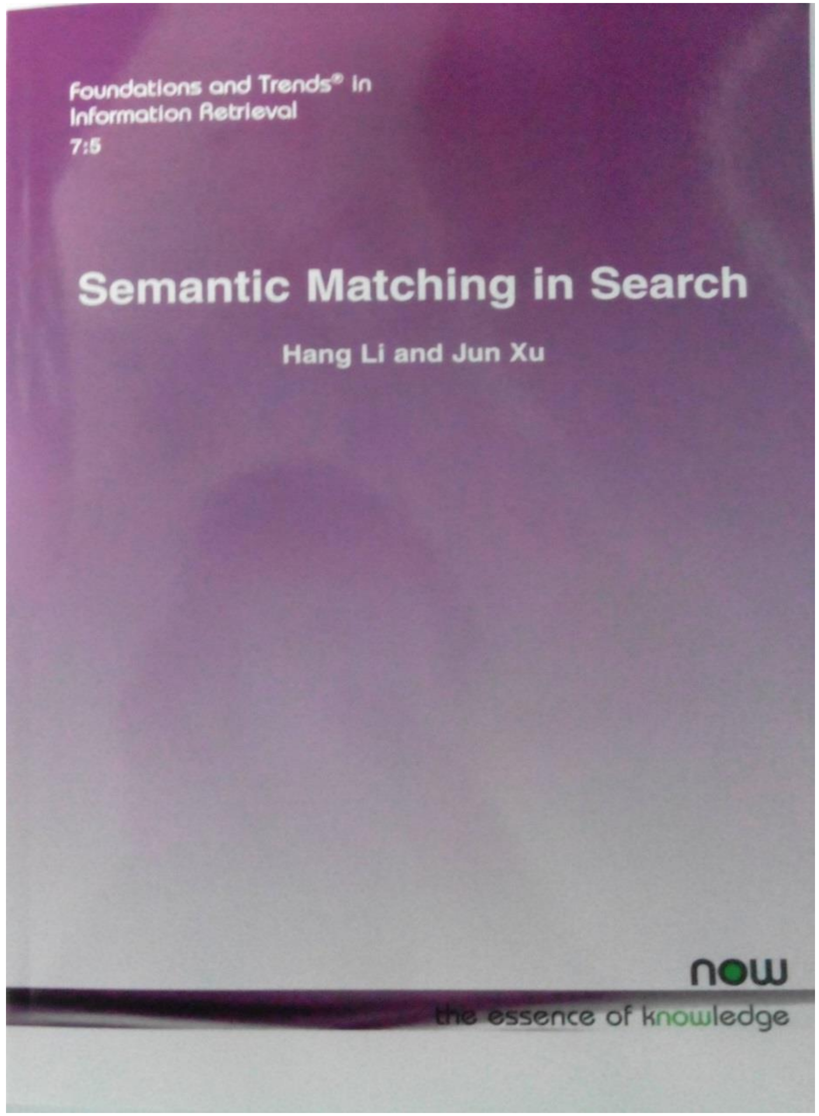
- Matching by query reformulation
- Matching with term dependency model
- Matching with translation model
- Matching with topic model
- Matching with latent space model

Characteristics of Approaches

	model	training data	complexity of learning
Query	query	search log	small
Dependency	query-document	relevance	small
Translation	query-document	click-through	small
Topic	document	document	high
Latent	query-document	click-through	high

Open Problems

- Topic drift: language is by nature synonymous and polysemous
- Scalability: e.g., topic model and latent space model needs large scale computing environment
- Missing information in training data: for rare queries and documents
- More NLP techniques is needed: for long queries and NLP queries
- Evaluation measures: Current approaches has limitation



http://www.nowpublishers.com/articles/foundations-and-trends-in-information-retrieval/INR-035
http://www.hangli-hl.com/uploads/3/1/6/8/3168008/ml_for_match-step2.pdf

Q & A

Thank you!

junxu@ict.ac.cn